elasticsearch.

Who we are

Uri Boness

- Co-founder SearchWorkings
- @uboness

Shay Banon

- Founder of ElasticSearch
- @kimchy

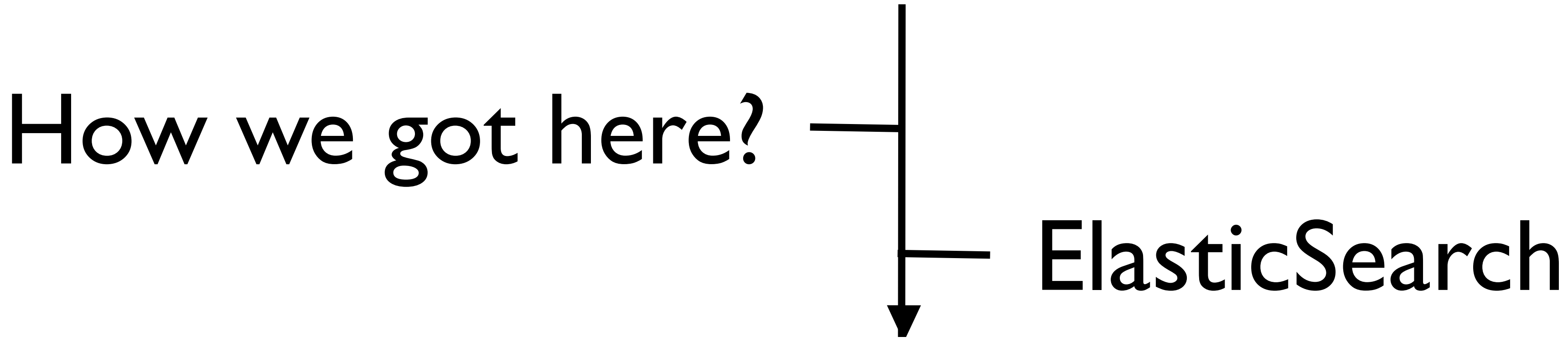
The Next Hour

↓

The Next Hour

How we got here?

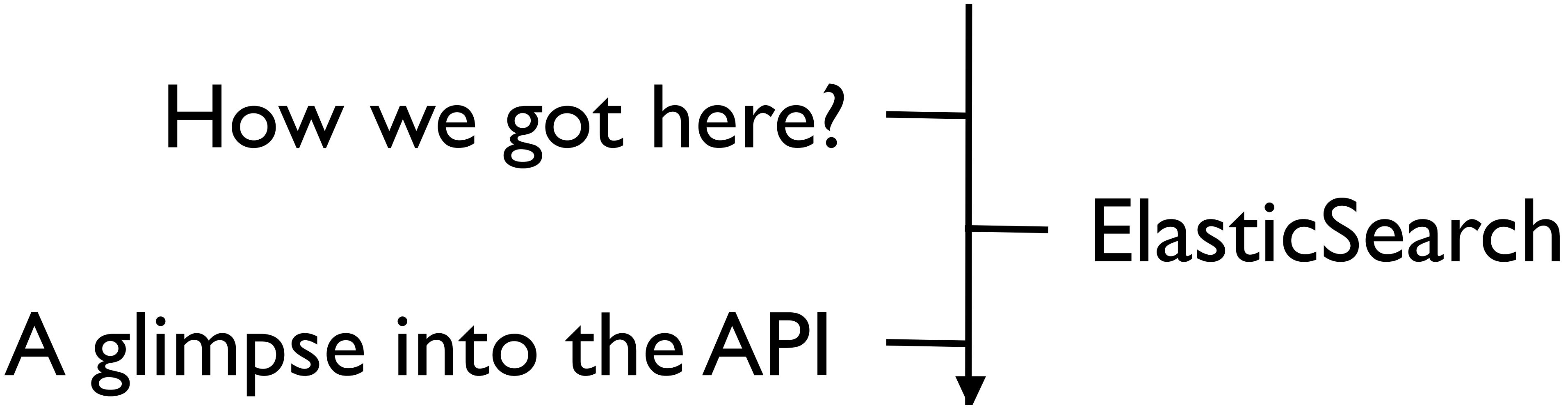
The Next Hour



How we got here?

ElasticSearch

The Next Hour

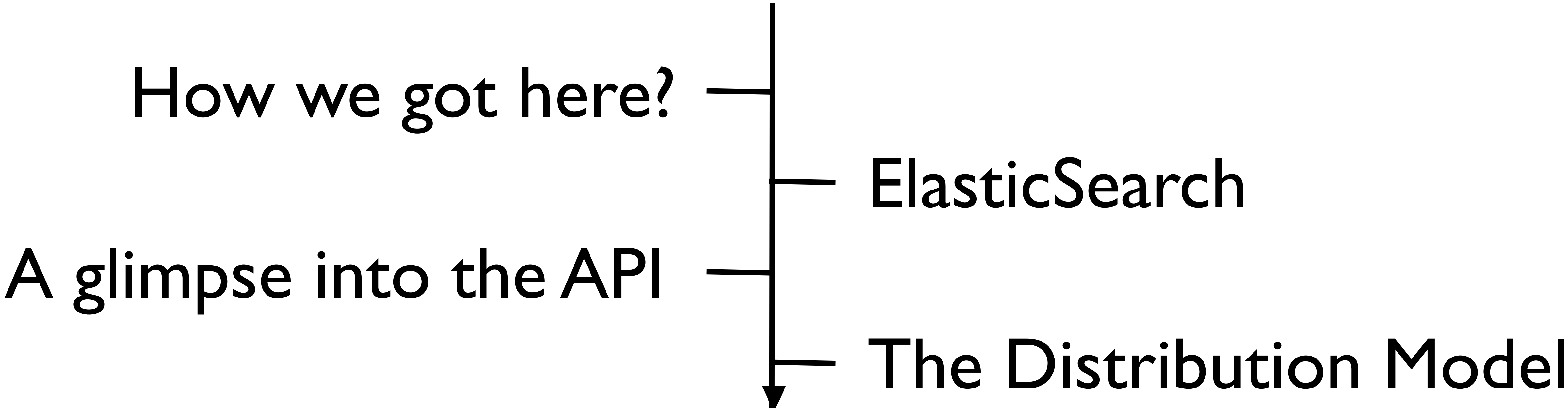


How we got here?

ElasticSearch

A glimpse into the API

The Next Hour



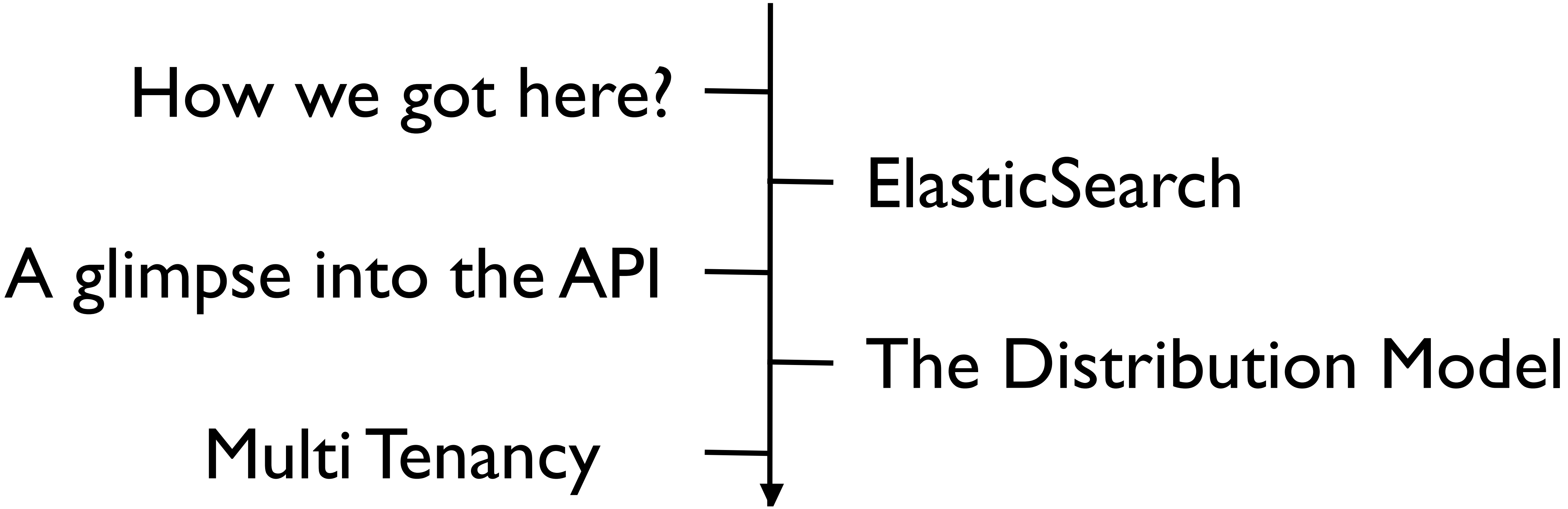
How we got here?

ElasticSearch

A glimpse into the API

The Distribution Model

The Next Hour



How we got here?

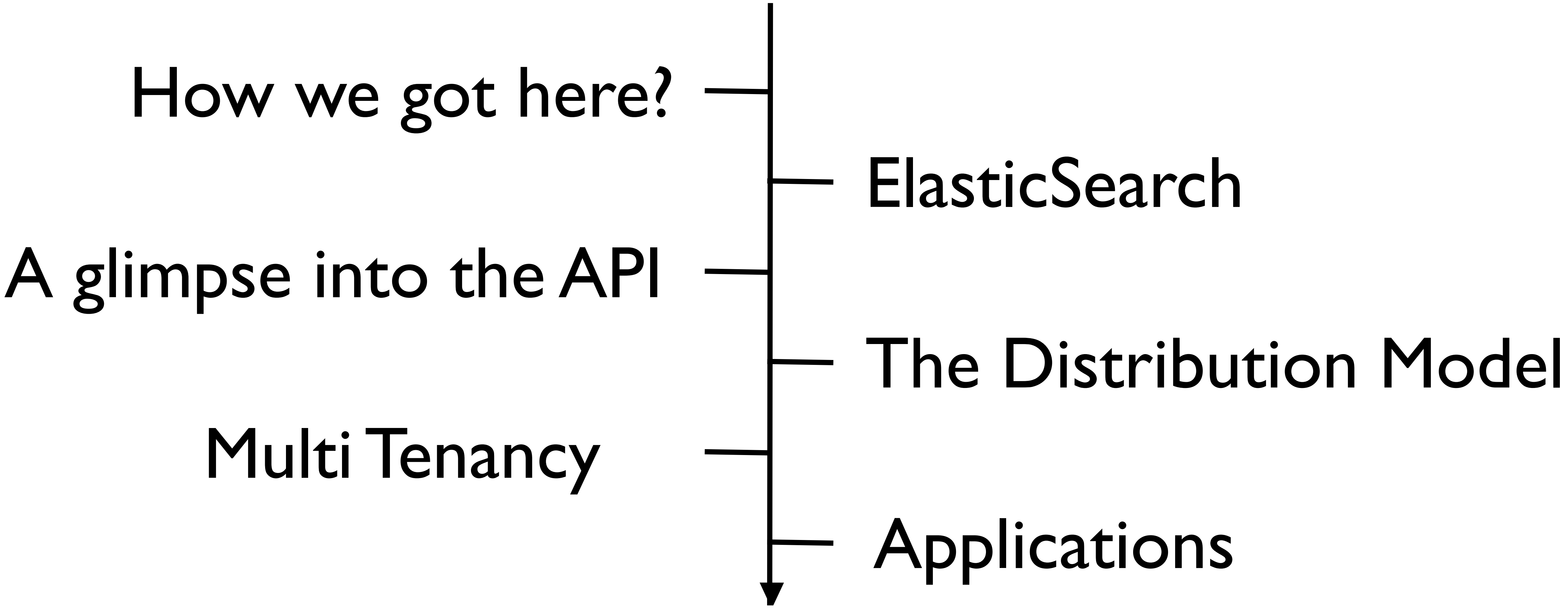
ElasticSearch

A glimpse into the API

The Distribution Model

Multi Tenancy

The Next Hour



How we got here?

ElasticSearch

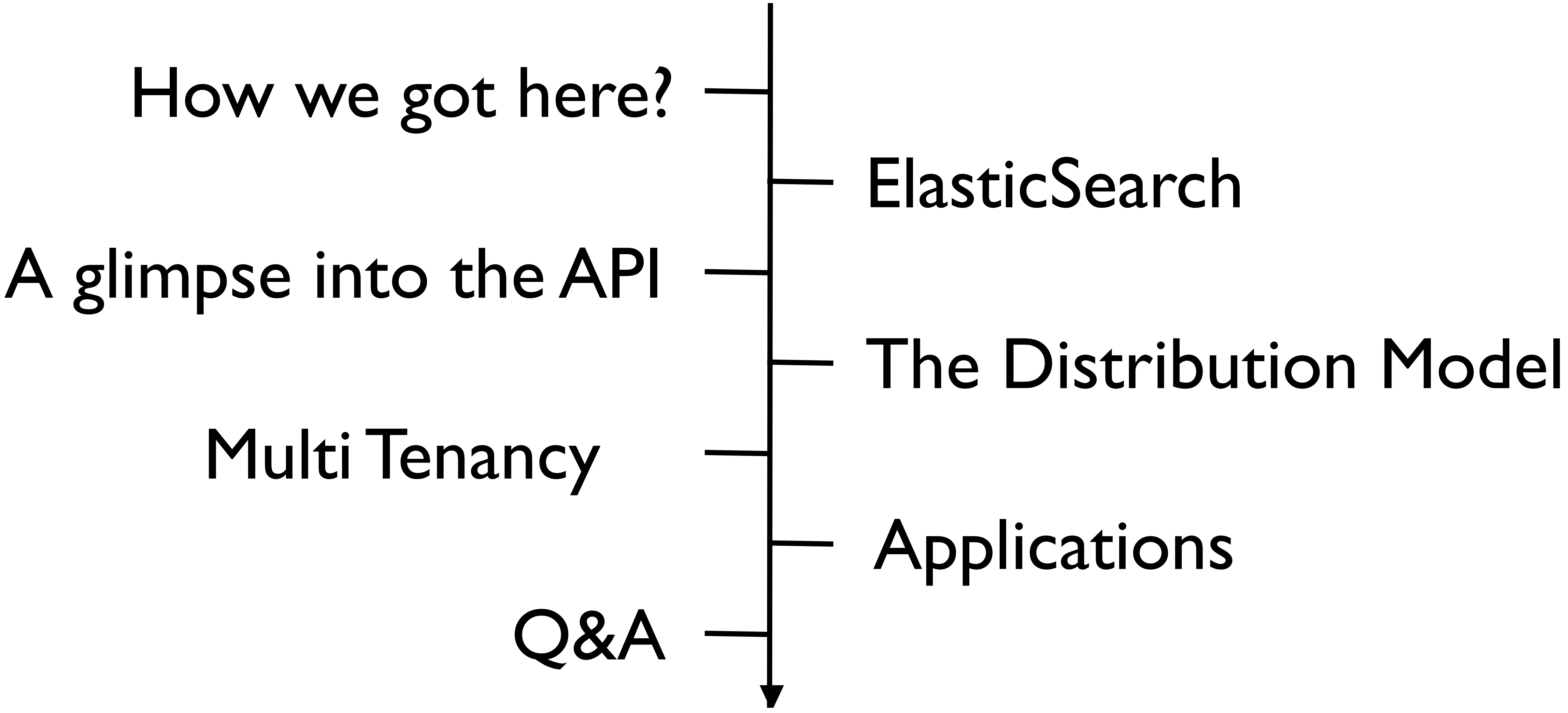
A glimpse into the API

The Distribution Model

Multi Tenancy

Applications

The Next Hour



How we got here?

ElasticSearch

A glimpse into the API

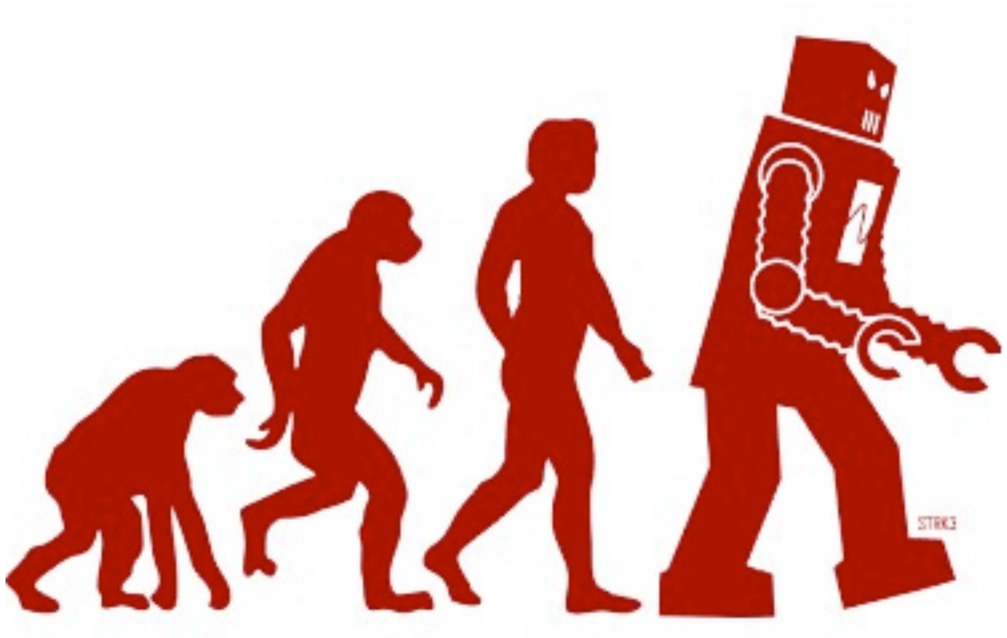
The Distribution Model

Multi Tenancy

Applications

Q&A

How we got here?



Search - Past

- Traditional "Enterprise" Search
- Federated Search
- Monolithic "do it all" Systems
  - Connectors
  - Document convertors/processors
  - (Enterprise) Security
  - oh yeah... and Search

Search - Present

- Findablility First
  - Free text, faceting, ranking, etc...
- Other top concerns:
  - Scale
  - Maintenance
  - Real time
- Cloud
- DevOps are programmers
  - Chef, Puppet, Whirr, Script languages

Search - Future

- All about data accessibility & insight
- Real time-ness
- Scale (Big Data)
    - Store
    - Query/Search
    - Analyze
- Familiar & consistent data model and infrastructure

ElasticSearch

- A highly scalable and distributed search engine
- Built on top of Lucene
- Platform & Environment agnostic
- Founded & mainly developed by Shay Banon
- Vibrant community
- Production ready & mature

ElasticSearch
API

API Design

- Simplicity
- Natural
- Platform friendliness
- Human friendliness
- Consistency
- Extensibility

API Design

- Simplicity
- Natural
- Platform friendliness
- Human friendliness
- Consistency
- Extensibility

REST

REST API Design

vs.

api for all

- Why?
  - Consistency
  - Runtime maintainability
  - DevOps are programmers

- What?
  - Data (Index, Update, Delete, Search)
  - Management & Maintenance
  - Monitoring

Dictionary

Dictionary

- Documents & Fields
- Document Type
- Index
- Node
- Cluster

Design Decisions

- Default format: JSON
- Zero Conf. Policy
  - System provides defaults for everything
  - Enables overriding all defaults

Data API

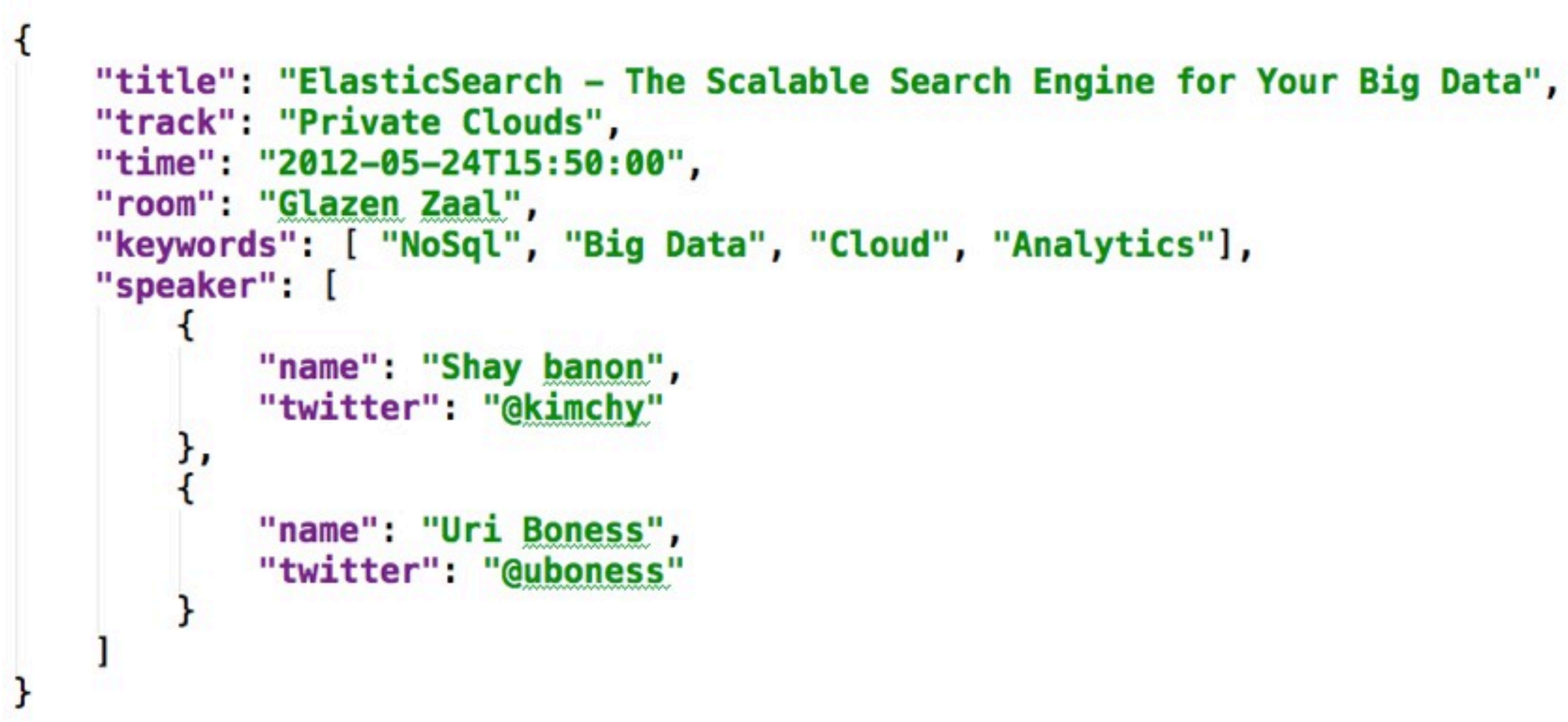
- Index
- Search
  - Query DSL
- Update, Delete

Index

- Index
- Delete (by id / query)
- Update
- Bulk API (not covered here)

Indexing - Add

```
PUT http://localhost:9200/goto-adam/session/1
```



```
{
    "title": "ElasticSearch - The Scalable Search Engine for Your Big Data",
    "track": "Private Clouds",
    "time": "2012-05-24T15:50:00",
    "room": "Glazen Zaal",
    "keywords": [ "NoSql", "Big Data", "Cloud", "Analytics"],
    "speaker": [
        {
            "name": "Shay banon",
            "twitter": "@kimchy"
        },
        {
            "name": "Uri Boness",
            "twitter": "@uboness"
        }
    ]
}
```

Indexing - Add

```
PUT http://localhost:9200/goto-adam/session/1
```

Indexing - Add

```
PUT http://localhost:9200/goto-adam/session/1
```

index

Indexing - Add

```
PUT http://localhost:9200/goto-adam/session/1
```

index

type

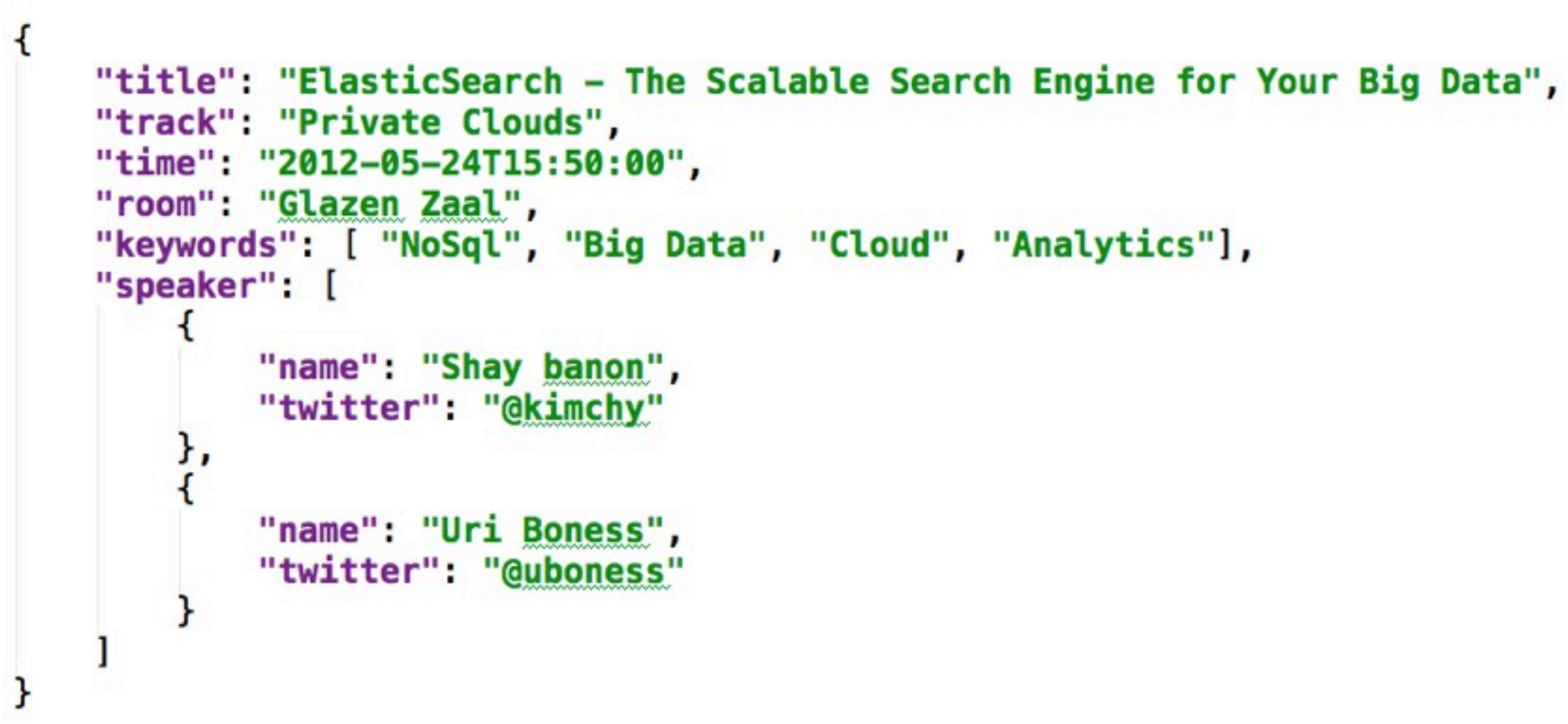
Indexing - Add

```
PUT http://localhost:9200/goto-adam/session/1
                          index        type   id
```

Indexing - Add

PUT `http://localhost:9200/goto-adam/session/1`

index type id

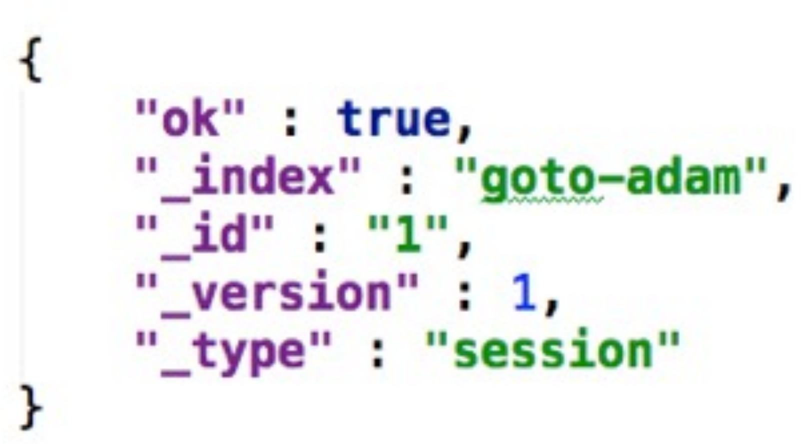


```
{
    "title": "ElasticSearch - The Scalable Search Engine for Your Big Data",
    "track": "Private Clouds",
    "time": "2012-05-24T15:50:00",
    "room": "Glazen Zaal",
    "keywords": [ "NoSql", "Big Data", "Cloud", "Analytics"],
    "speaker": [
        {
            "name": "Shay banon",
            "twitter": "@kimchy"
        },
        {
            "name": "Uri Boness",
            "twitter": "@uboness"
        }
    ]
}
```

Indexing - Add

PUT `http://localhost:9200/goto-adam/session/1`

index type id



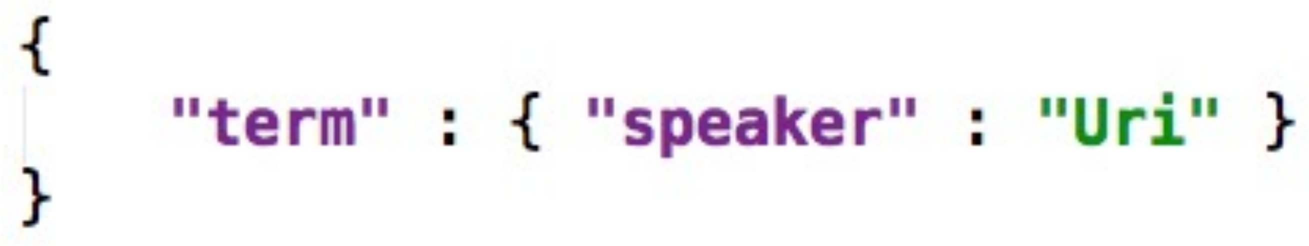
```
{
    "ok" : true,
    "_index" : "goto-adam",
    "_id" : "1",
    "_version" : 1,
    "_type" : "session"
}
```

Indexing - Delete

```
DELETE http://localhost:9200/goto-adam/session/1
```

OR

```
DELETE http://localhost:9200/goto-adam/session/_query
```

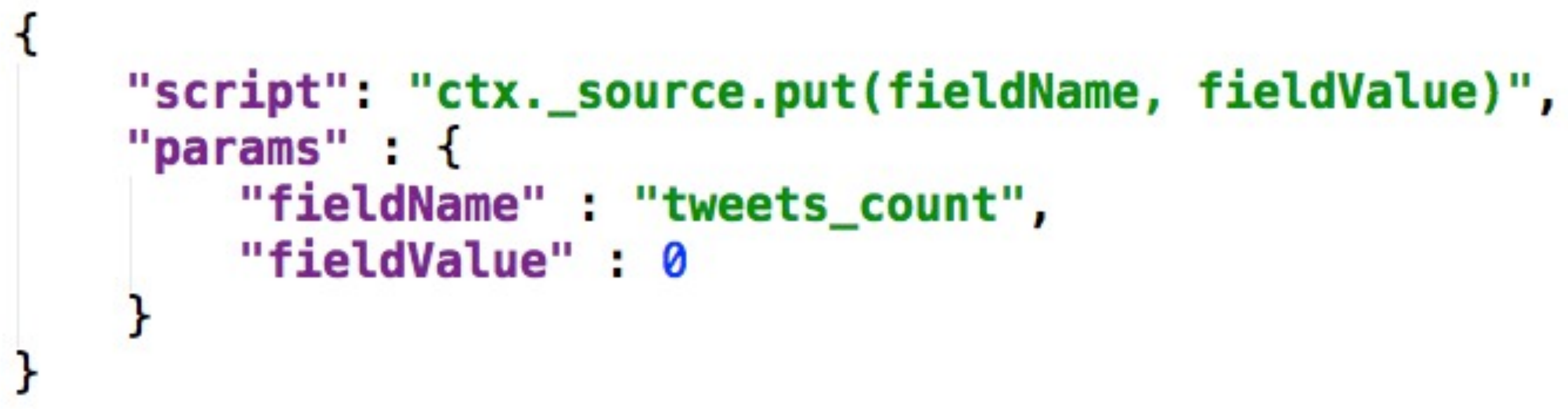


```
{
    "term" : { "speaker" : "Uri" }
}
```

Indexing - Update

Let’s track the number tweets mentioning this talk:

```
POST http://localhost:9200/goto-adam/session/1/_update
```

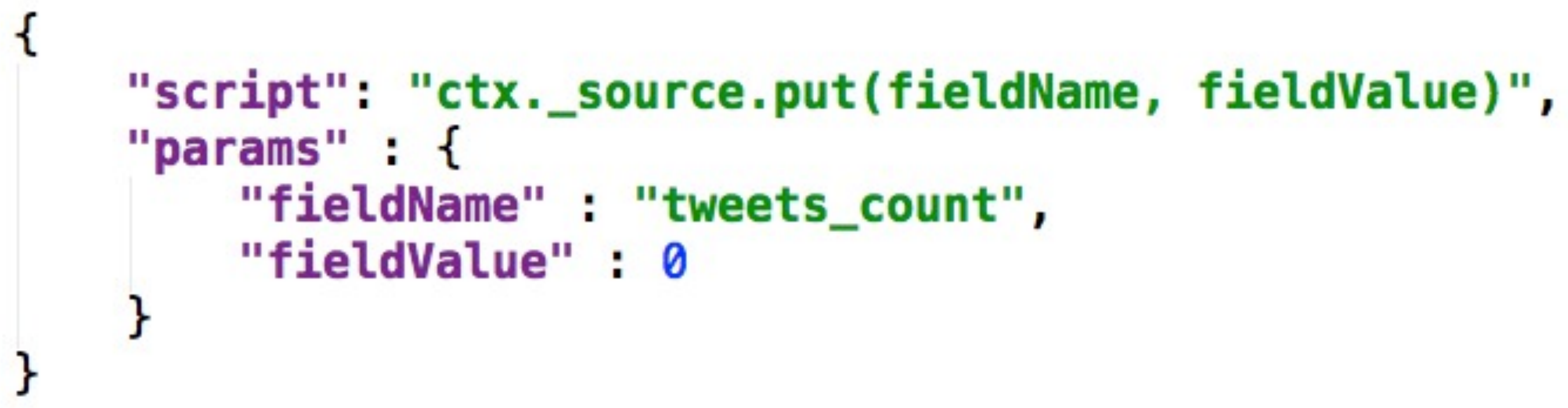


```
{
    "script": "ctx._source.put(fieldName, fieldValue)",
    "params" : {
        "fieldName" : "tweets_count",
        "fieldValue" : 0
    }
}
```

Indexing - Update

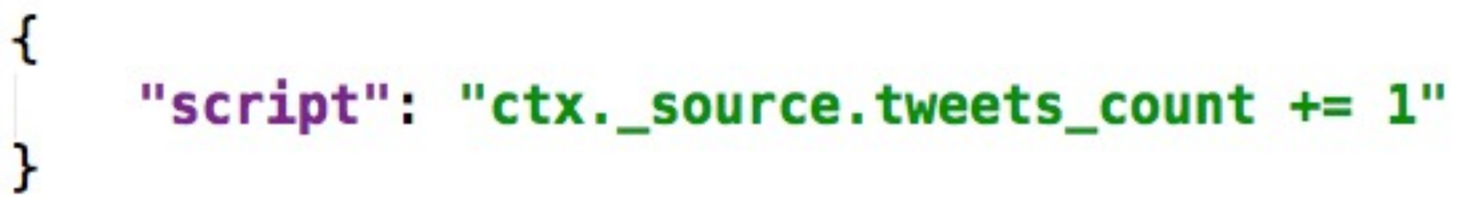
Let’s track the number tweets mentioning this talk:

```
POST http://localhost:9200/goto-adam/session/1/_update
```



```
{
    "script": "ctx._source.put(fieldName, fieldValue)",
    "params" : {
        "fieldName" : "tweets_count",
        "fieldValue" : 0
    }
}
```

That’s better... from now on we just update the count



```
{
    "script": "ctx._source.tweets_count += 1"
}
```

Search

- Query DSL
- Simple query
- filtered query
- facets (terms & date histogram)
- Other supported search features

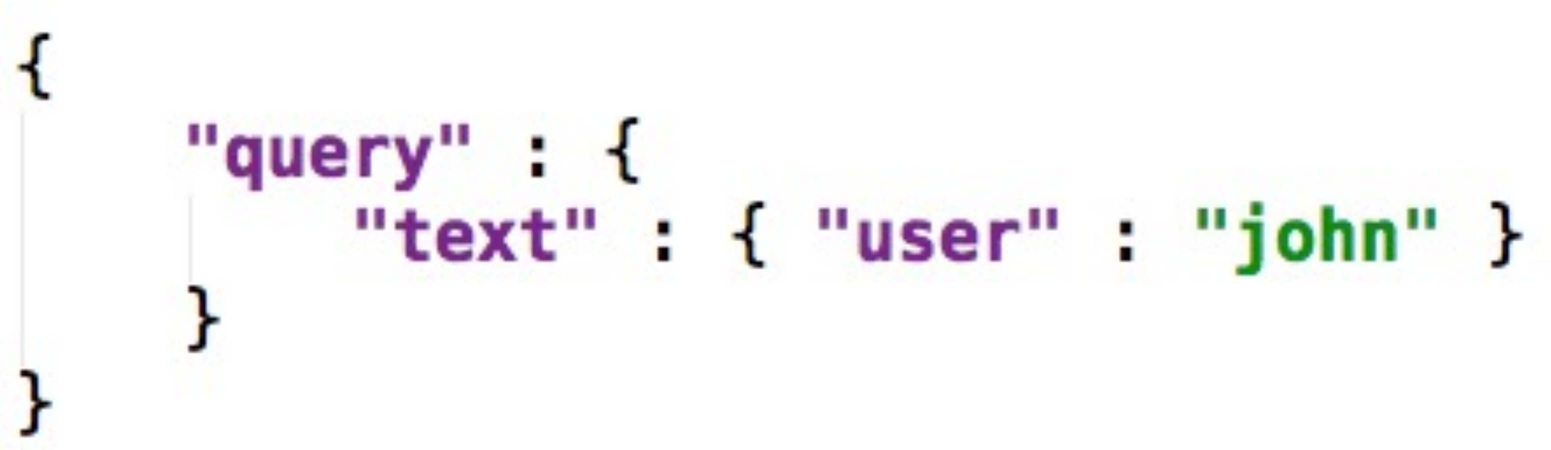
Query DSL

- Programming language friendly
- Tool friendly
- Self explanatory
- Fully supports all Lucene search constructs
  - All Lucene query types and filters
  - Additional query types (e.g. Geo, Parent/Child, Nested, and more)
- Easily extensible
  - Plug-in your own query types with their own custom DSL

Queries

Basic Query

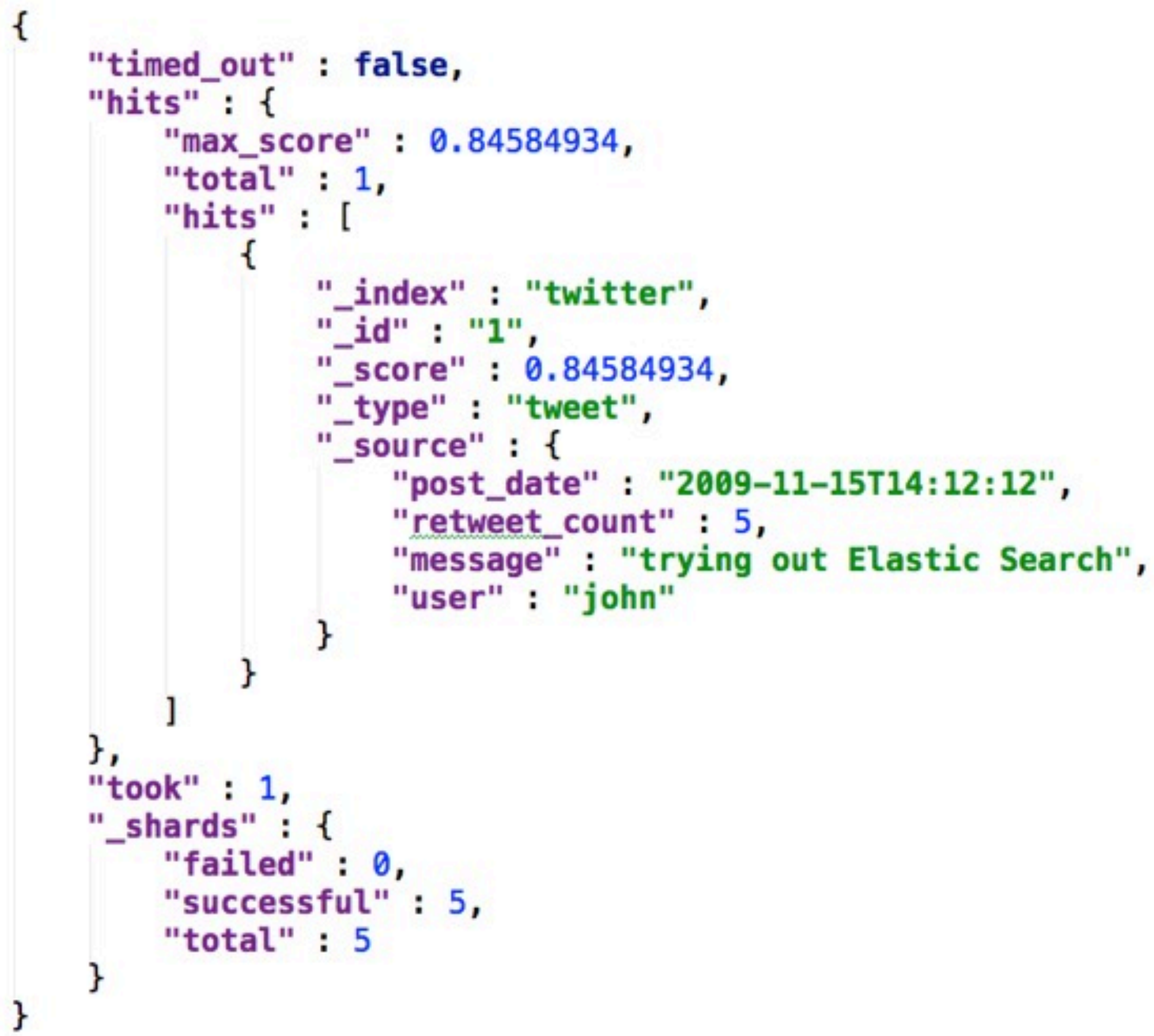
POST http://localhost:9200/twitter/tweet/_search



```
{
    "query" : {
        "text" : { "user" : "john" }
    }
}
```

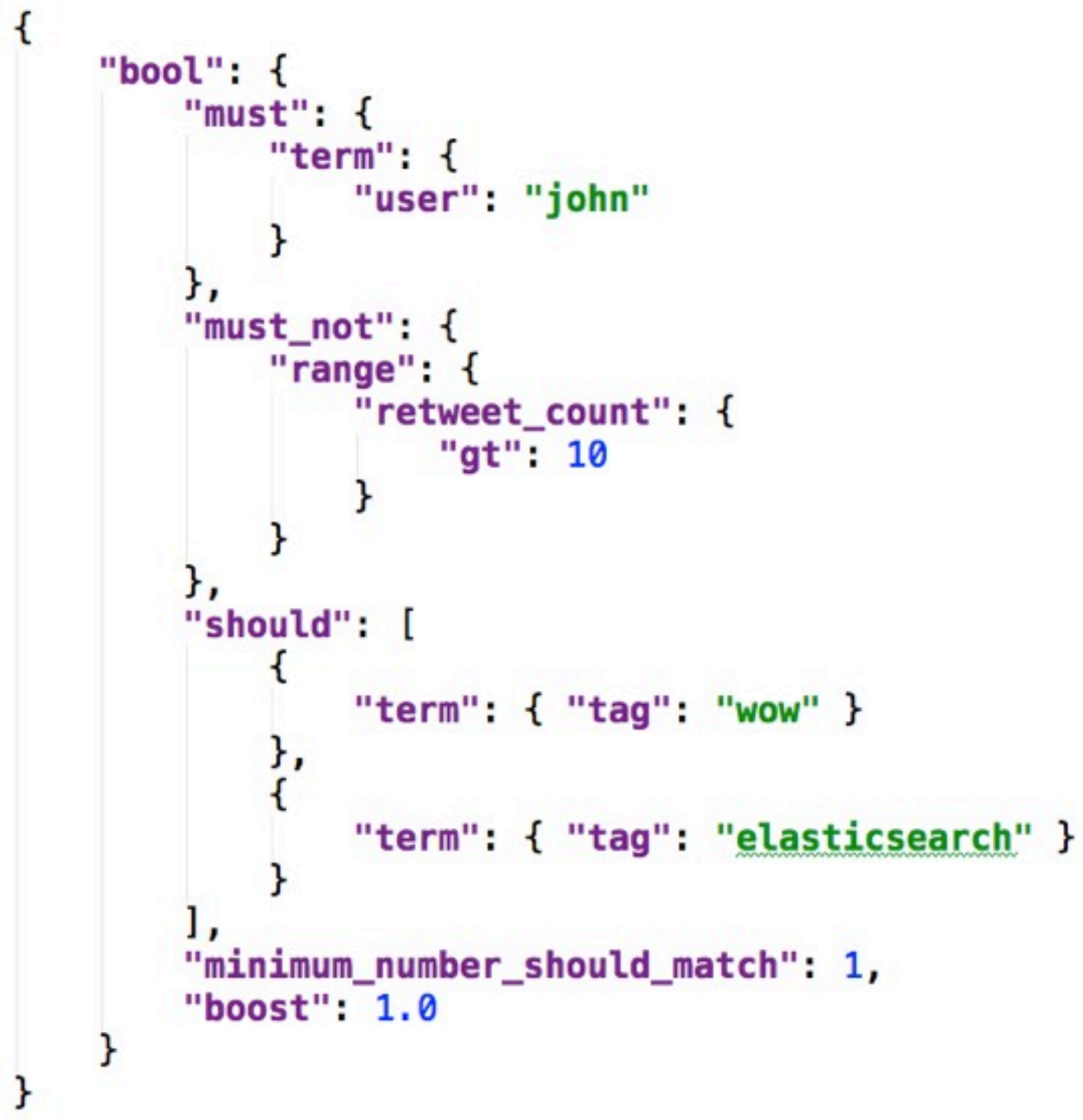
Basic Query

POST http://localhost:9200/twitter/tweet/_search



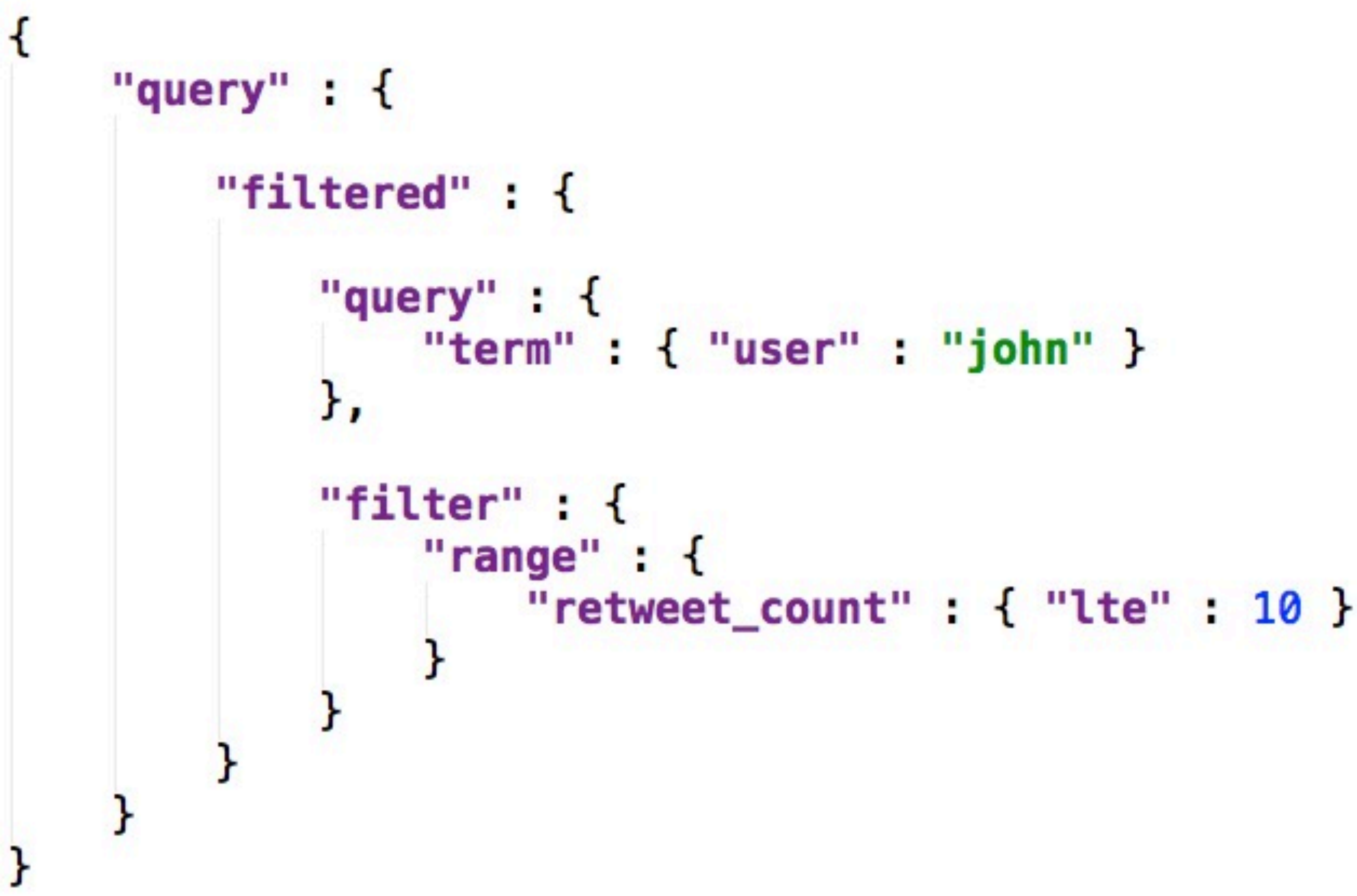
```
{
    "timed_out" : false,
    "hits" : {
        "max_score" : 0.84584934,
        "total" : 1,
        "hits" : [
            {
                "_index" : "twitter",
                "_id" : "1",
                "_score" : 0.84584934,
                "_type" : "tweet",
                "_source" : {
                    "post_date" : "2009-11-15T14:12:12",
                    "retweet_count" : 5,
                    "message" : "trying out Elastic Search",
                    "user" : "john"
                }
            }
        ]
    },
    "took" : 1,
    "_shards" : {
        "failed" : 0,
        "successful" : 5,
        "total" : 5
    }
}
```

Rich Boolean Queries



```
{
    "bool": {
        "must": {
            "term": {
                "user": "john"
            }
        },
        "must_not": {
            "range": {
                "retweet_count": {
                    "gt": 10
                }
            }
        },
        "should": [
            {
                "term": { "tag": "wow" }
            },
            {
                "term": { "tag": "elasticsearch" }
            }
        ],
        "minimum_number_should_match": 1,
        "boost": 1.0
    }
}
```

Filtered Queries



```
{
    "query" : {

        "filtered" : {

            "query" : {
                "term" : { "user" : "john" }
            },

            "filter" : {
                "range" : {
                    "retweet_count" : { "lte" : 10 }
                }
            }
        }
    }
}
```

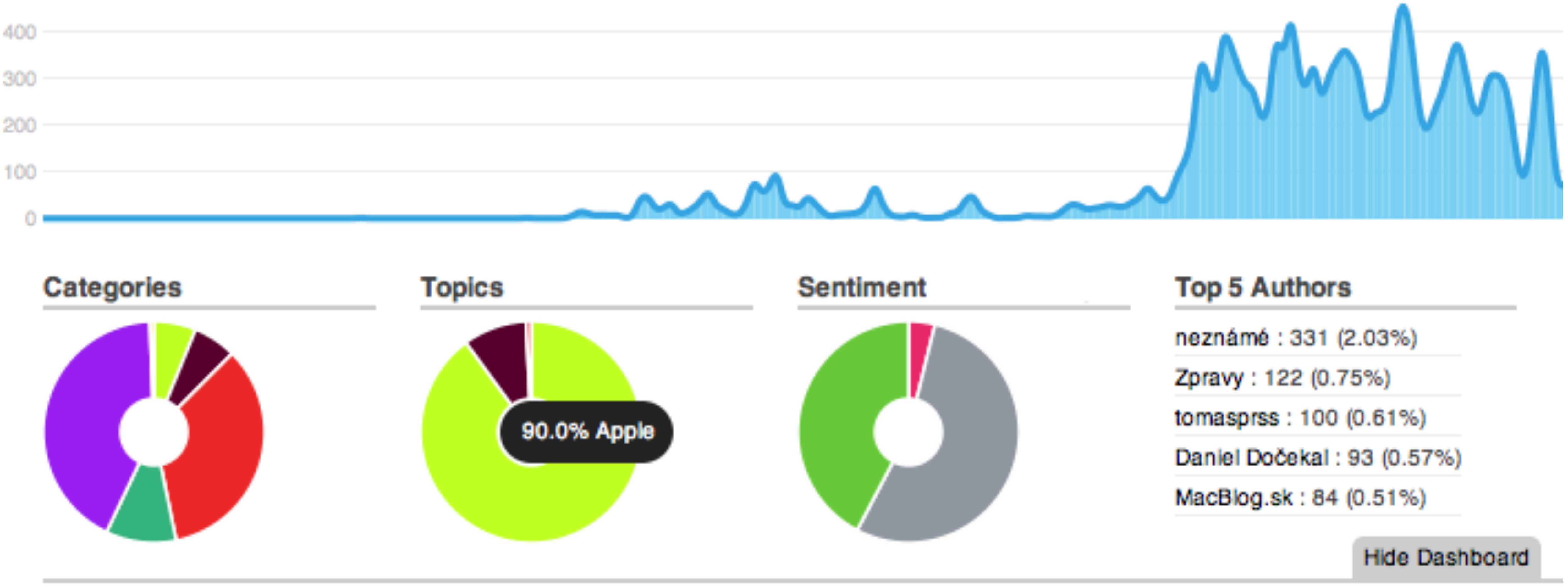
Query Types

- text, query_string, field
- term, range, prefix
- bool, dis_max
- custom_score, custom_filters_score
- ...

Filter Types

- term, range
- geo (distance, bbox, polygon)
- bool, and, or, not
- ...

Facets

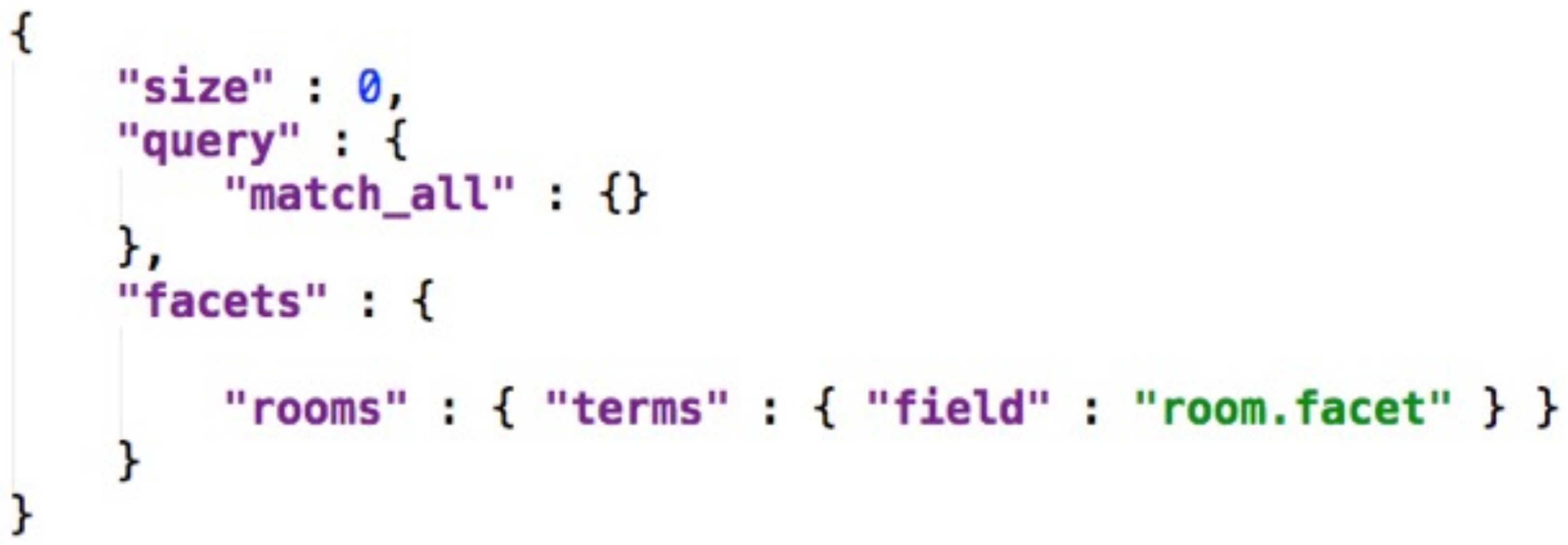


Top 5 Authors

neznámé : 331 (2.03%)
Zpravy : 122 (0.75%)
tomasprss : 100 (0.61%)
Daniel Dočekal : 93 (0.57%)
MacBlog.sk : 84 (0.51%)

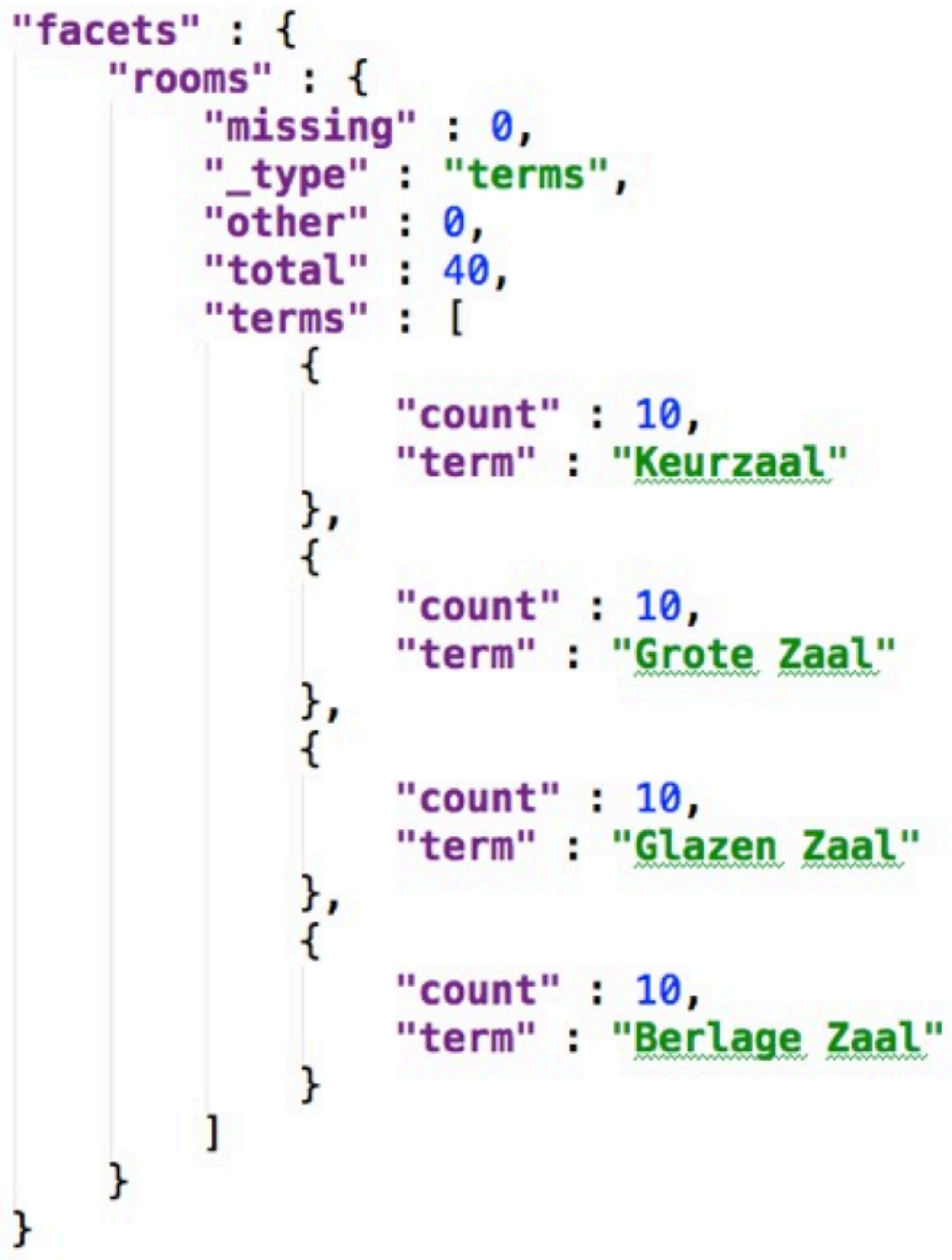
examples

Terms Facets



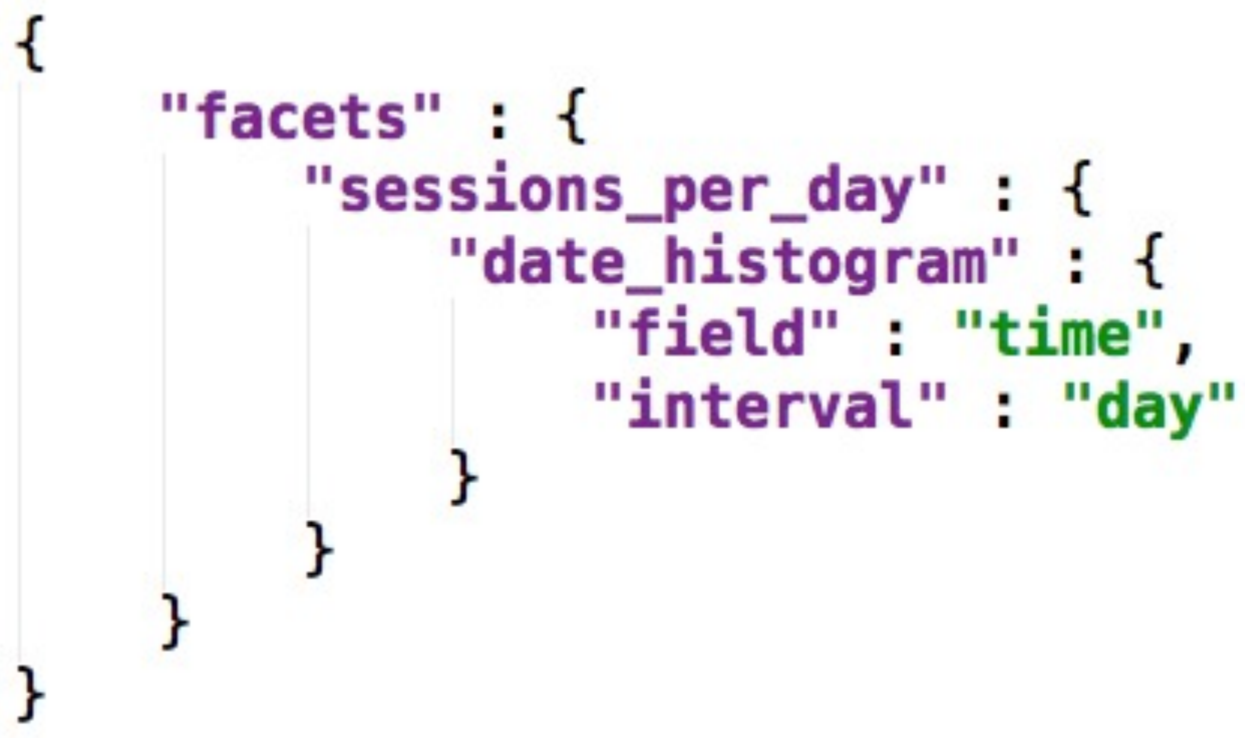
```
{
    "size" : 0,
    "query" : {
        "match_all" : {}
    },
    "facets" : {

        "rooms" : { "terms" : { "field" : "room.facet" } }
    }
}
```

Terms Facets



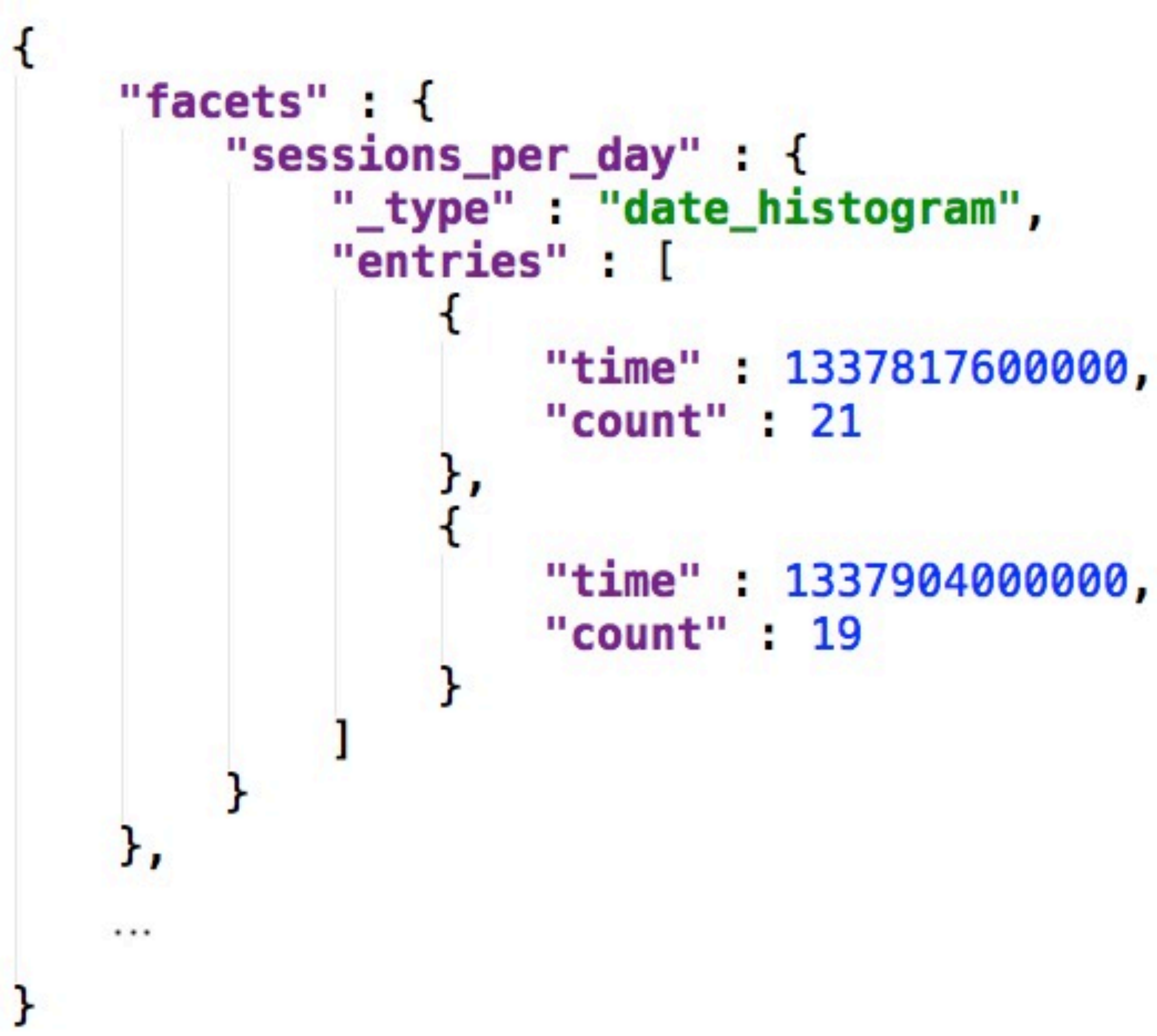
```
"facets" : {
    "rooms" : {
        "missing" : 0,
        "_type" : "terms",
        "other" : 0,
        "total" : 40,
        "terms" : [
            {
                "count" : 10,
                "term" : "Keurzaal"
            },
            {
                "count" : 10,
                "term" : "Grote Zaal"
            },
            {
                "count" : 10,
                "term" : "Glazen Zaal"
            },
            {
                "count" : 10,
                "term" : "Berlage Zaal"
            }
        ]
    }
}
```

Date Histogram



```
{
    "facets" : {
        "sessions_per_day" : {
            "date_histogram" : {
                "field" : "time",
                "interval" : "day"
            }
        }
    }
}
```

Date Histogram



```
{
    "facets" : {
        "sessions_per_day" : {
            "_type" : "date_histogram",
            "entries" : [
                {
                    "time" : 1337817600000,
                    "count" : 21
                },
                {
                    "time" : 1337904000000,
                    "count" : 19
                }
            ]
        }
    },
    ...
}
```

More Available Facets

- Histogram
- Statistical
- Terms Stats
- Range
- Geo Distance
- Filter

Other Features

- Pagination & Scrolling
- Sorting
- Highlighting
- Script Fields
- Realtime GET
- Multiple search types
- Min score filtering
- Named filters
- And much more...

Management API

- Indices
  - Create & Delete
    - Topology
    - Update Settings
  - Mapping
    - Put & Delete
  - Aliases & "Views"
  - Refresh, Flush, Optimize

- Cluster
  - Node shutdown
  - Update Settings

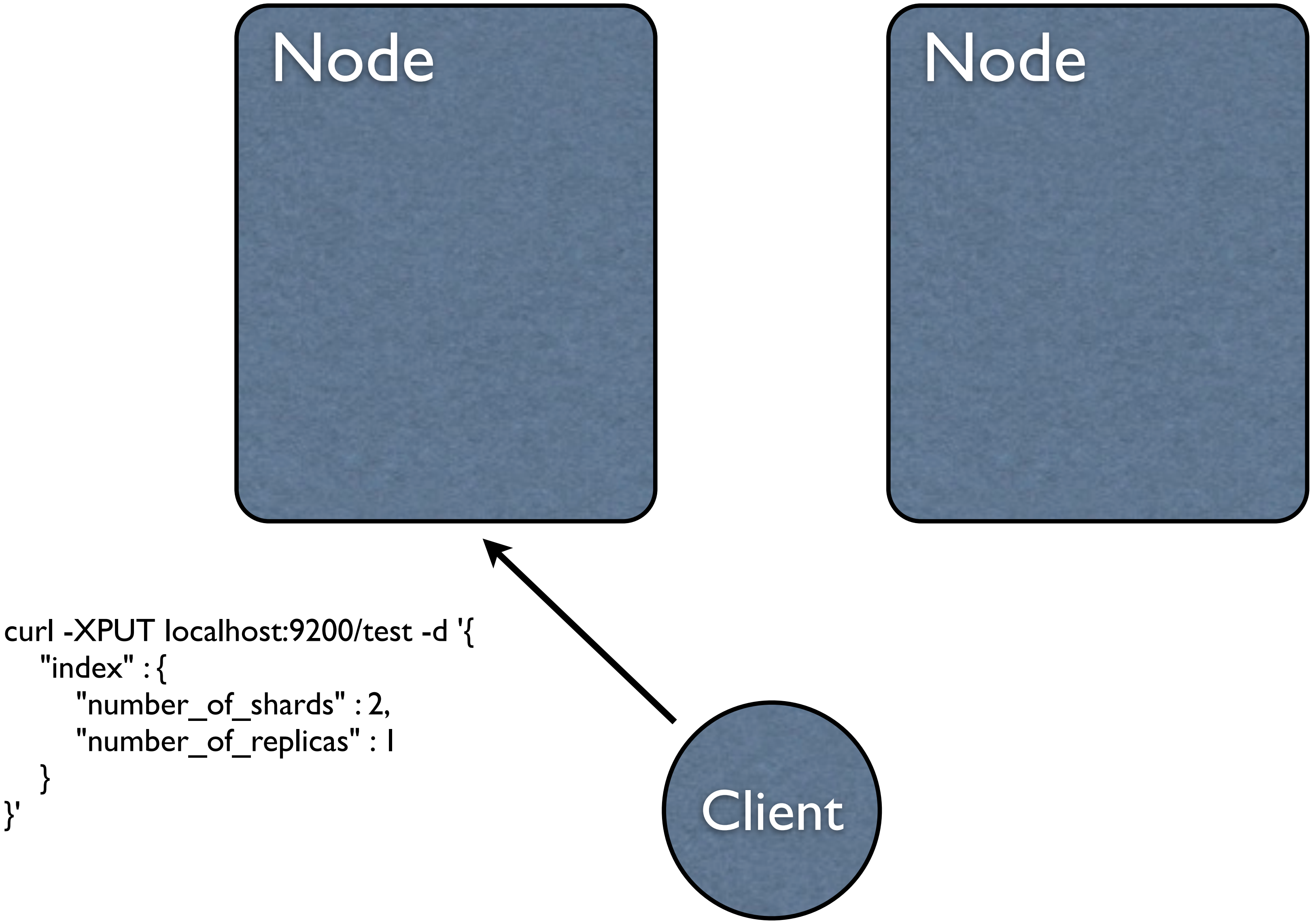
Monitoring API

- Index Level
  - State
  - Stats
  - Segments Info (Low level Lucene)

- Cluster Level
  - Health
  - State
  - Nodes stats

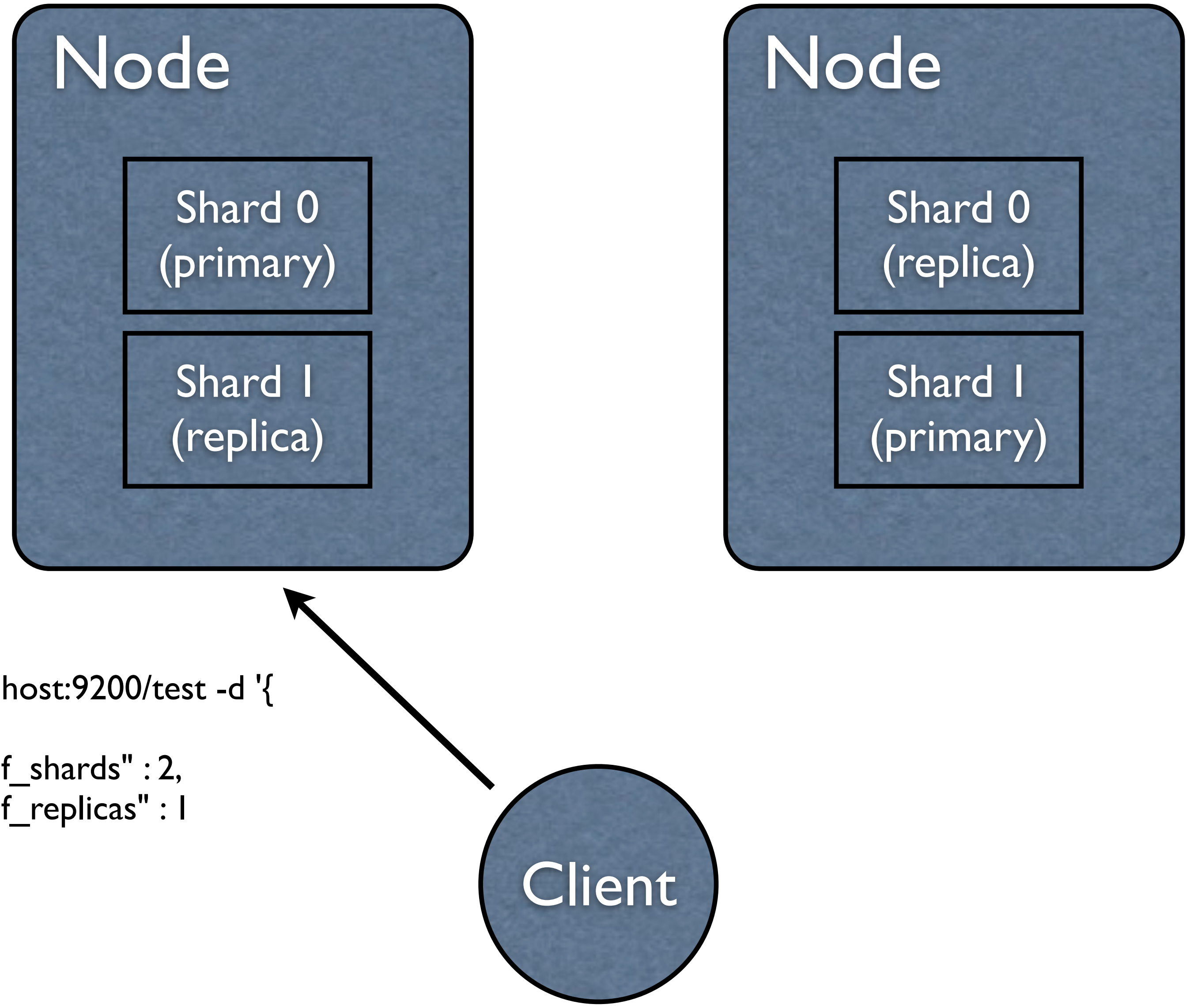
Distribution Model

index - shards and replicas



```
curl -XPUT localhost:9200/test -d '{
    "index" : {
        "number_of_shards" : 2,
        "number_of_replicas" : 1
    }
}'
```

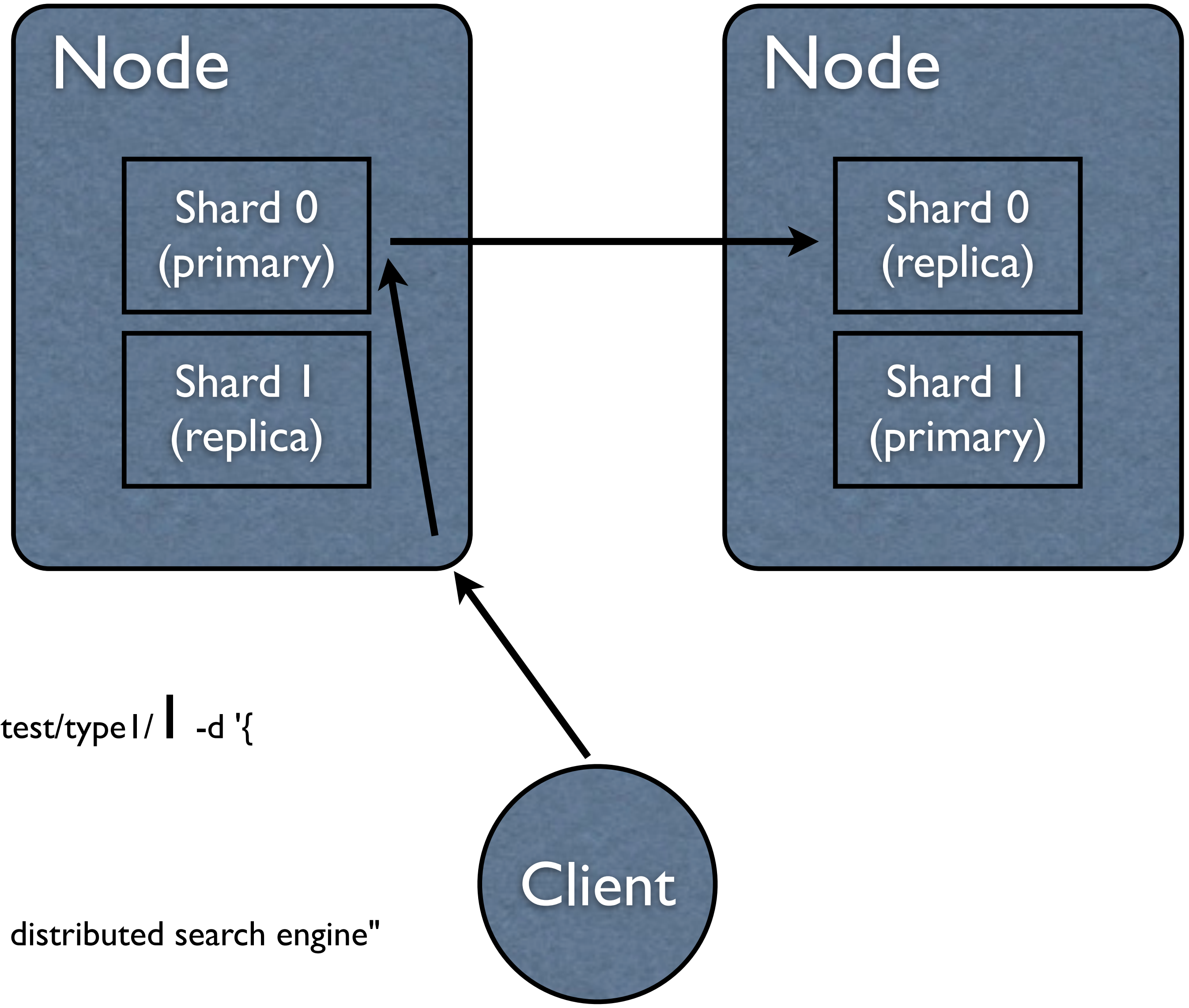
index - shards and replicas



```
curl -XPUT localhost:9200/test -d '{
    "index" : {
        "number_of_shards" : 2,
        "number_of_replicas" : 1
    }
}'
```

indexing - 1

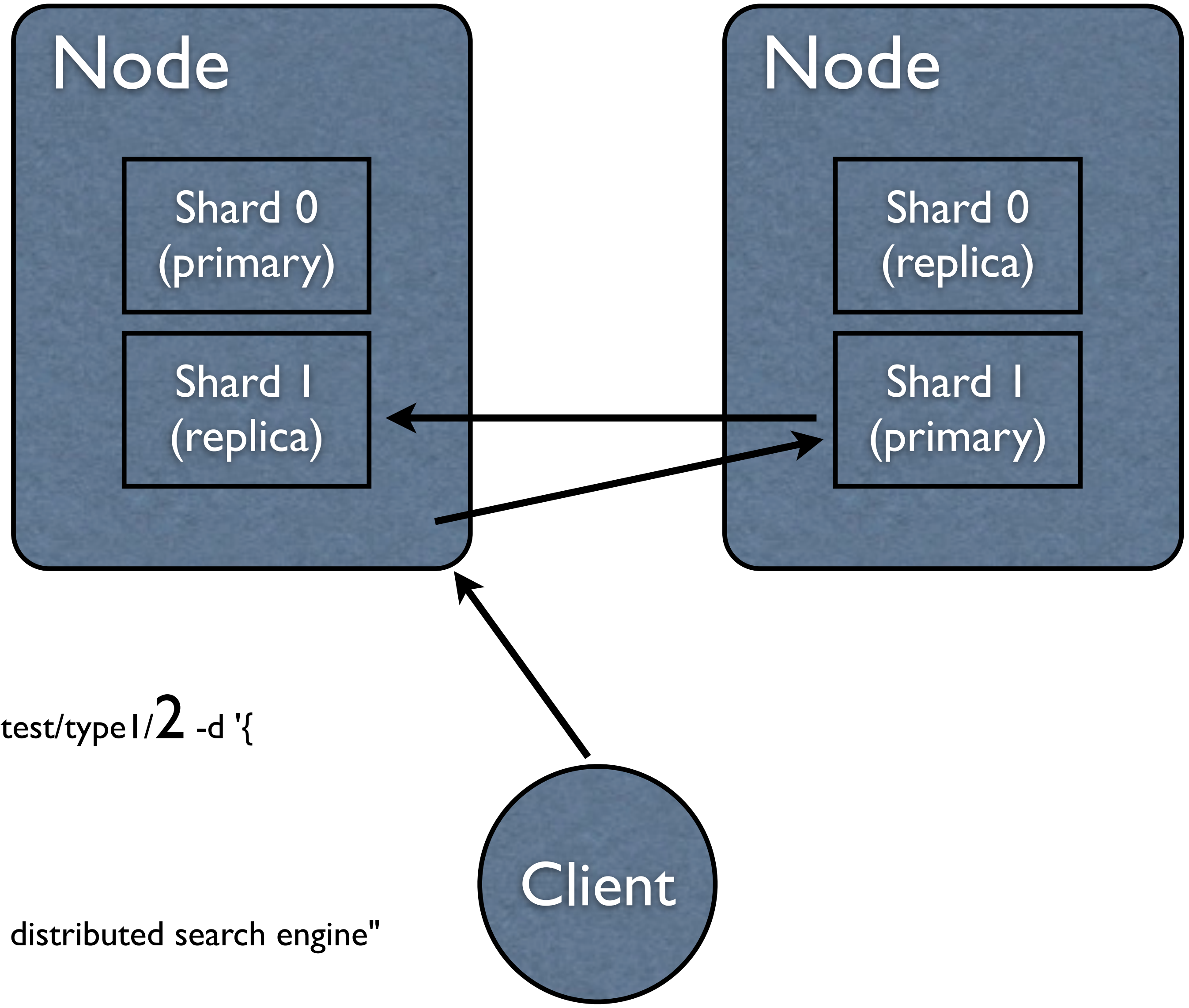
- Automatic sharding, push replication



```
curl -XPUT localhost:9200/test/type1/1 -d '{
    "name" : {
        "first" : "Shay",
        "last" : "Banon"
    } ,
    "title" : "ElasticSearch - A distributed search engine"
}'
```

indexing - 2

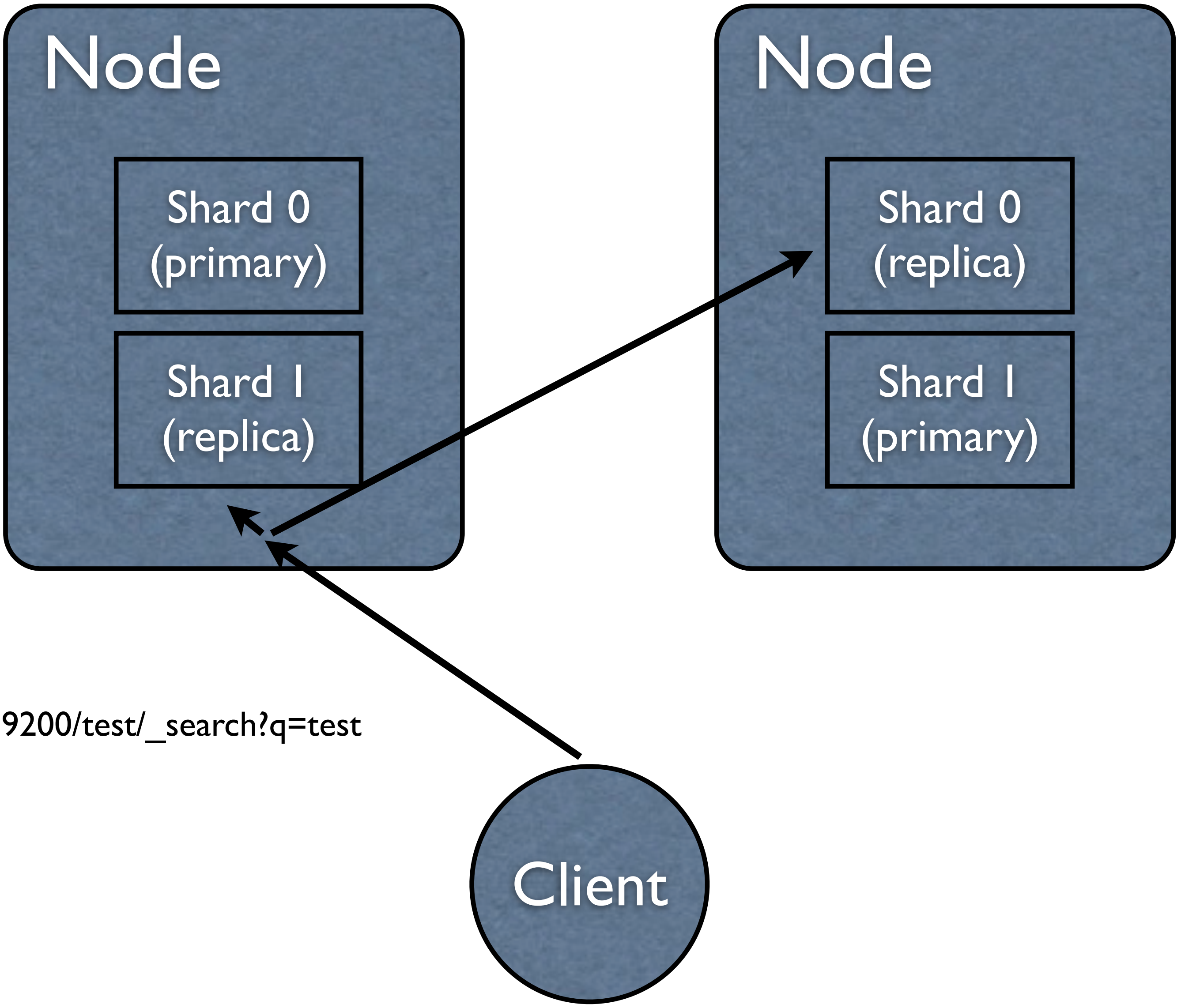
- Automatic request “redirection”



```
curl -XPUT localhost:9200/test/type1/2 -d '{
    "name" : {
        "first" : "Shay",
        "last" : "Banon"
    } ,
    "title" : "ElasticSearch - A distributed search engine"
}'
```

search - 1

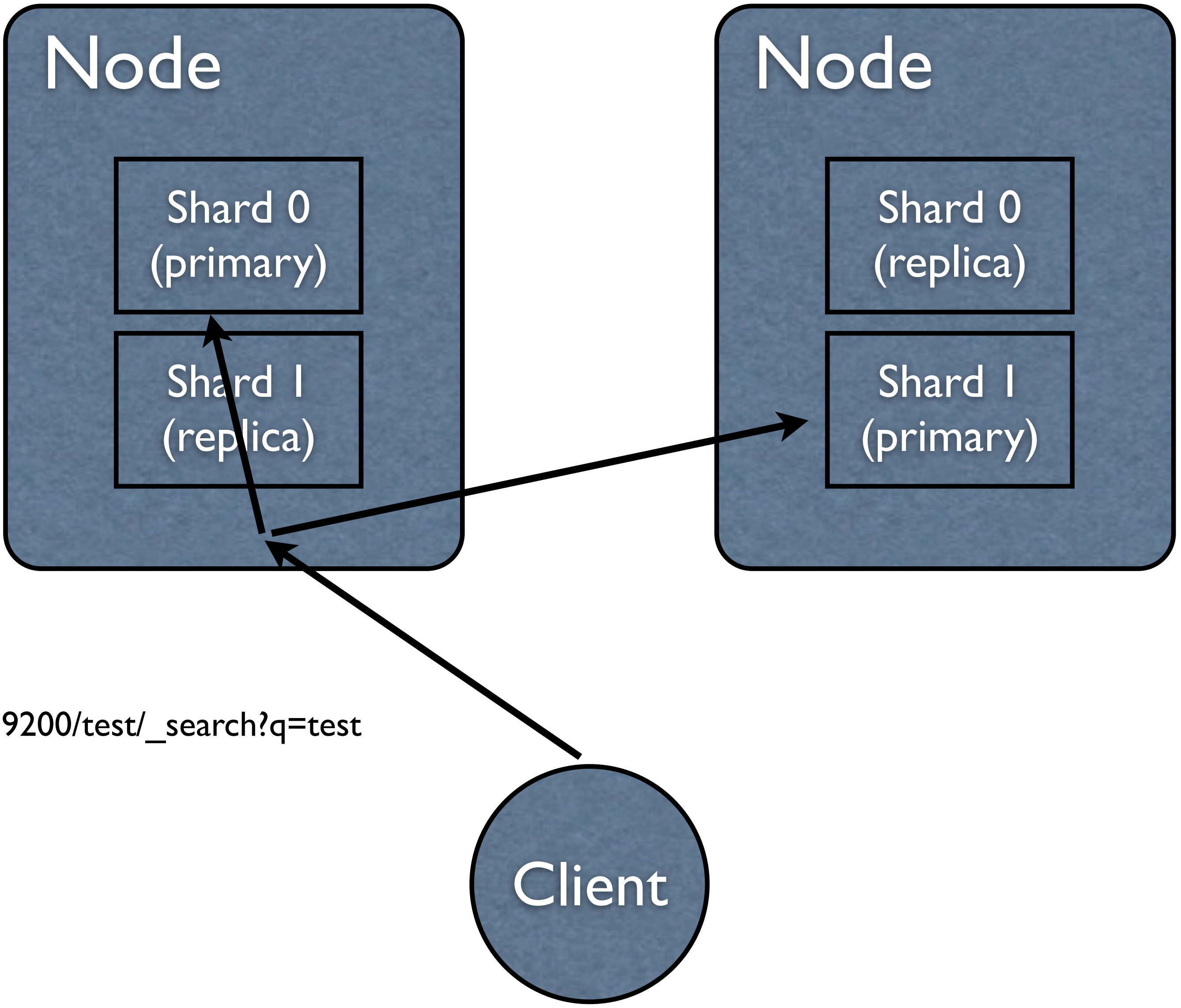
- Scatter / Gather search



curl -XPUT localhost:9200/test/_search?q=test

search - 2

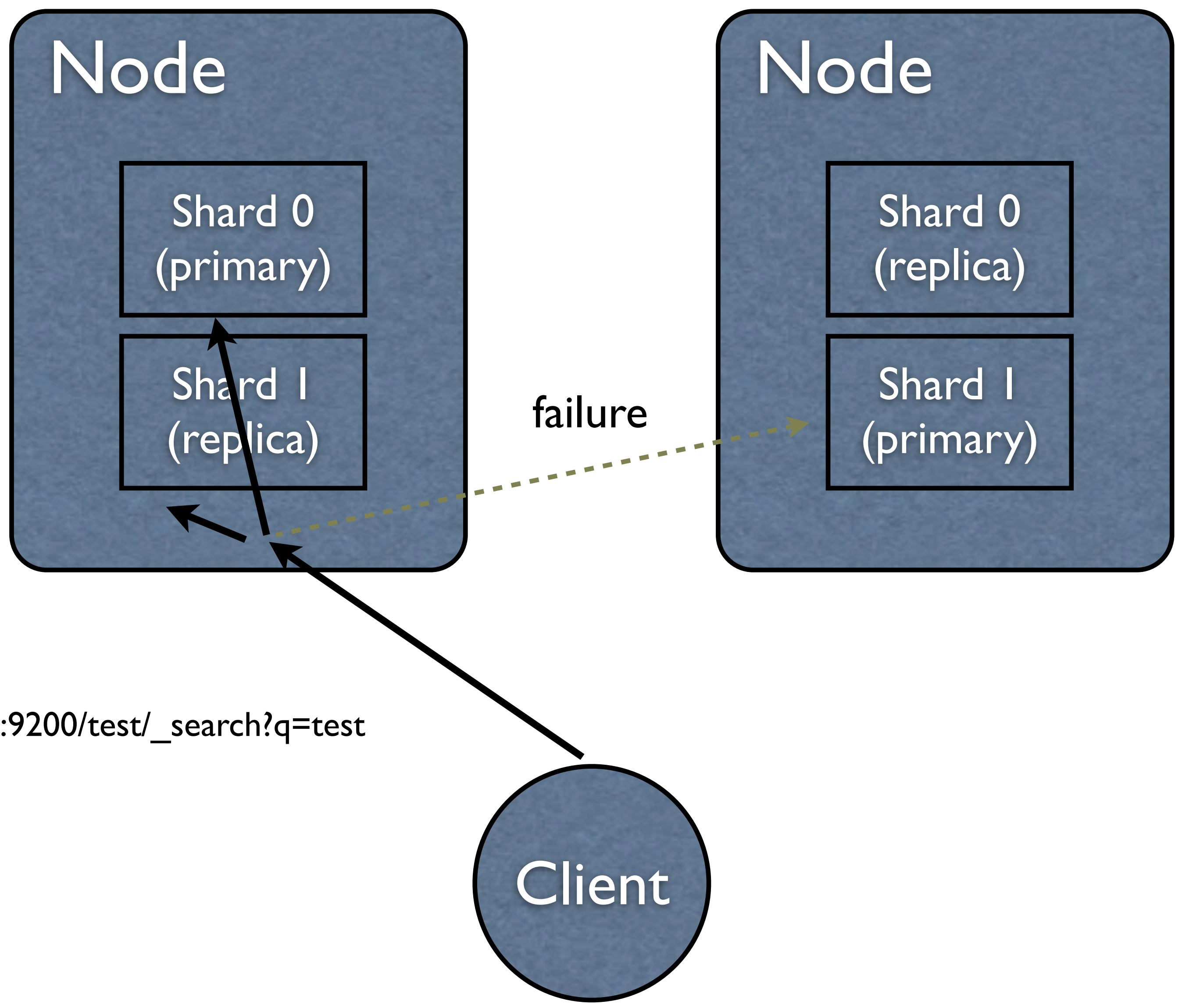
- Automatic balancing between replicas



curl -XPUT localhost:9200/test/_search?q=test

search - 3

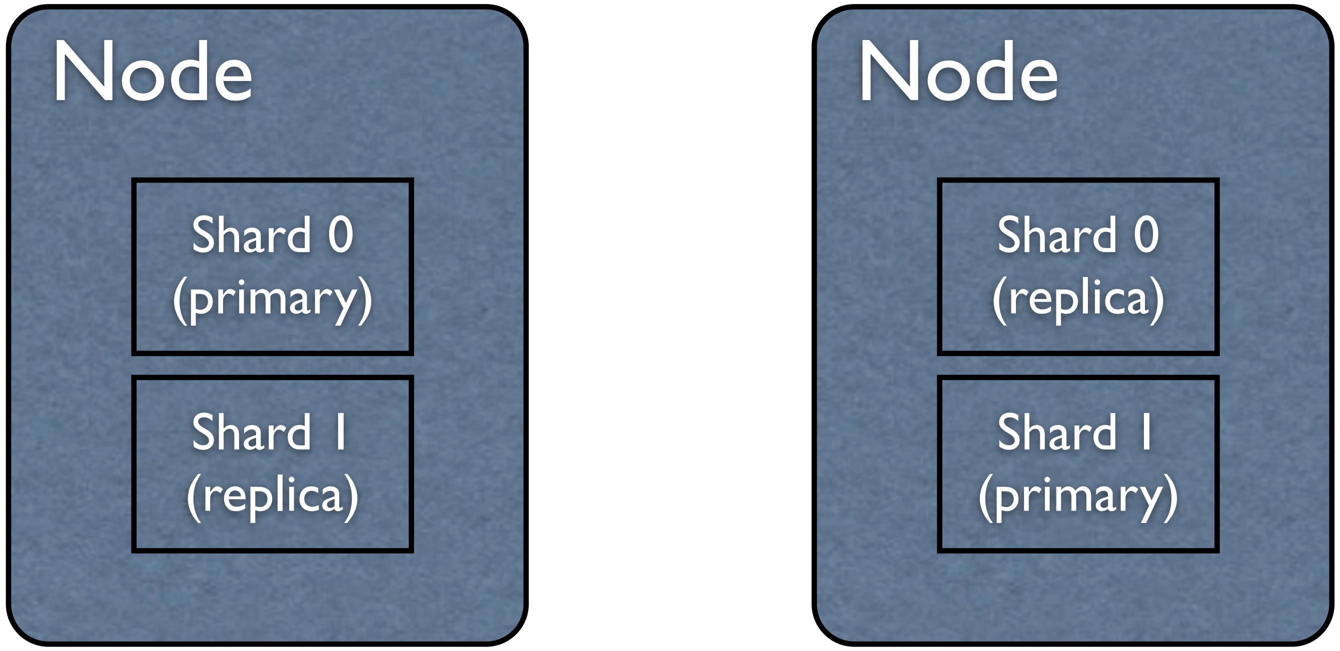
- Automatic failover



curl -XPUT localhost:9200/test/_search?q=test

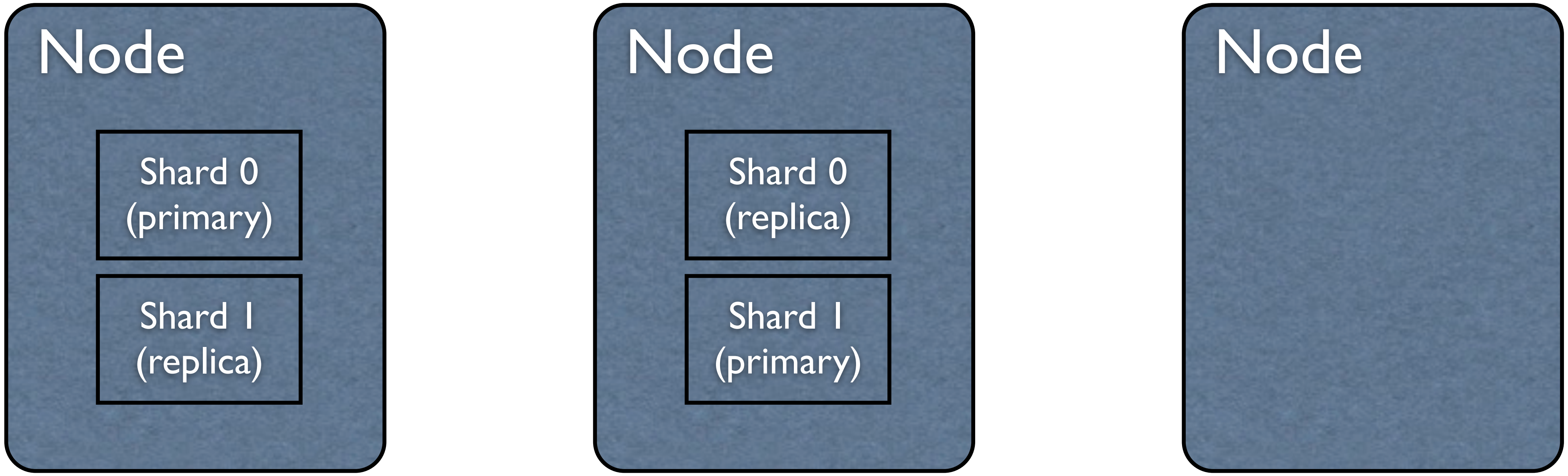
adding a node

- "Hot" relocation of shards to the new node



adding a node

- “Hot” relocation of shards to the new node



Node

Shard 0 (primary)

Shard 1 (replica)

Node

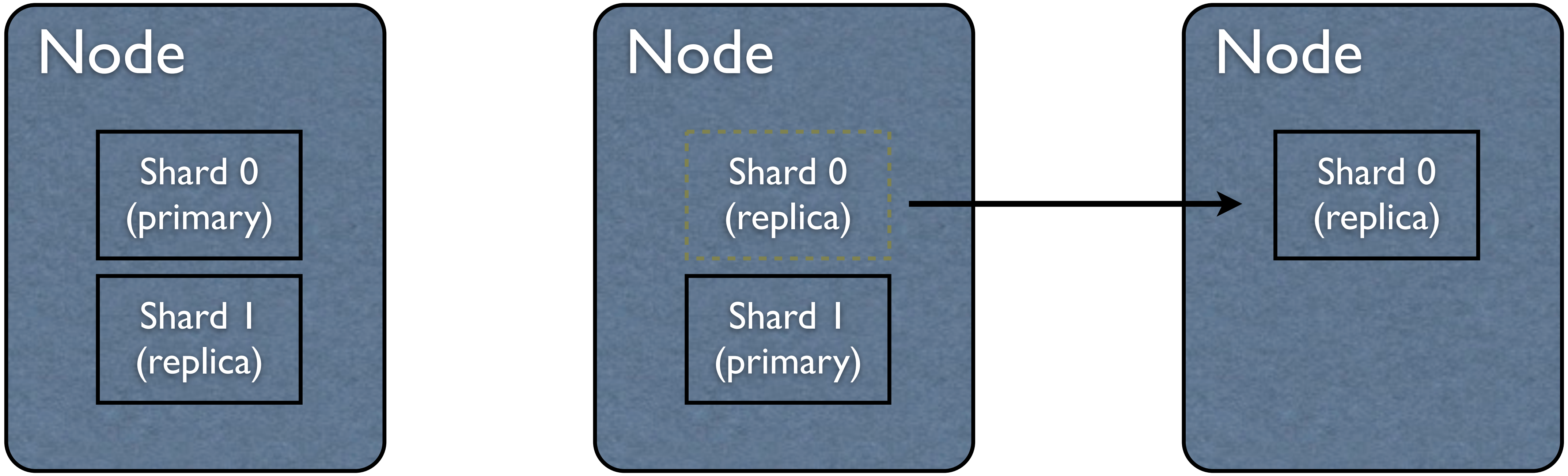
Shard 0 (replica)

Shard 1 (primary)

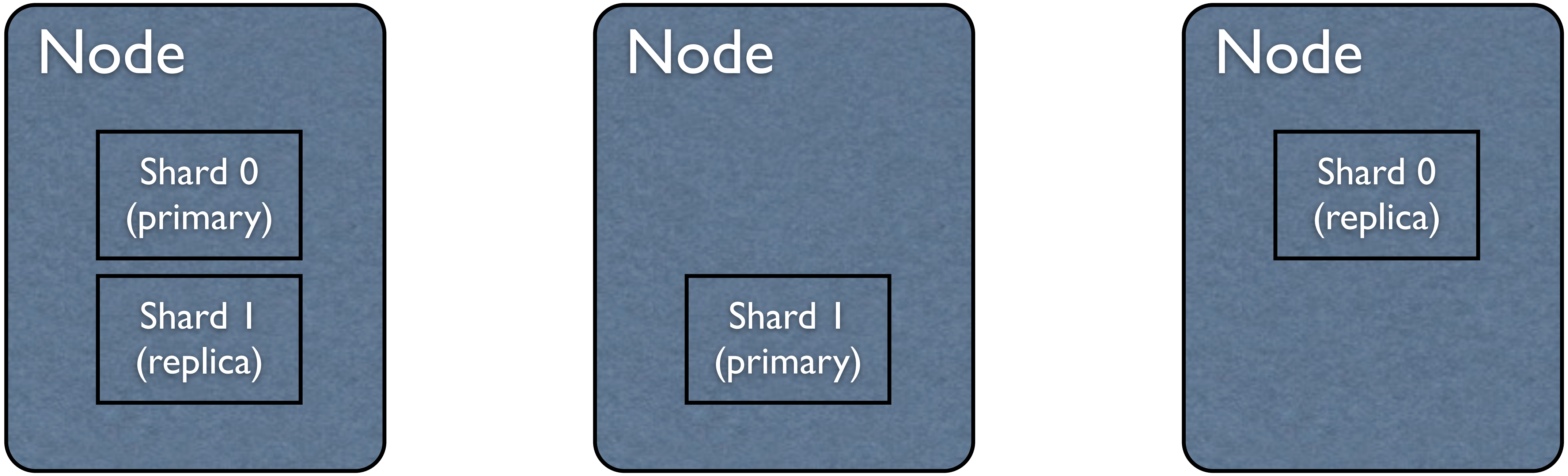
Node

adding a node

- “Hot” relocation of shards to the new node

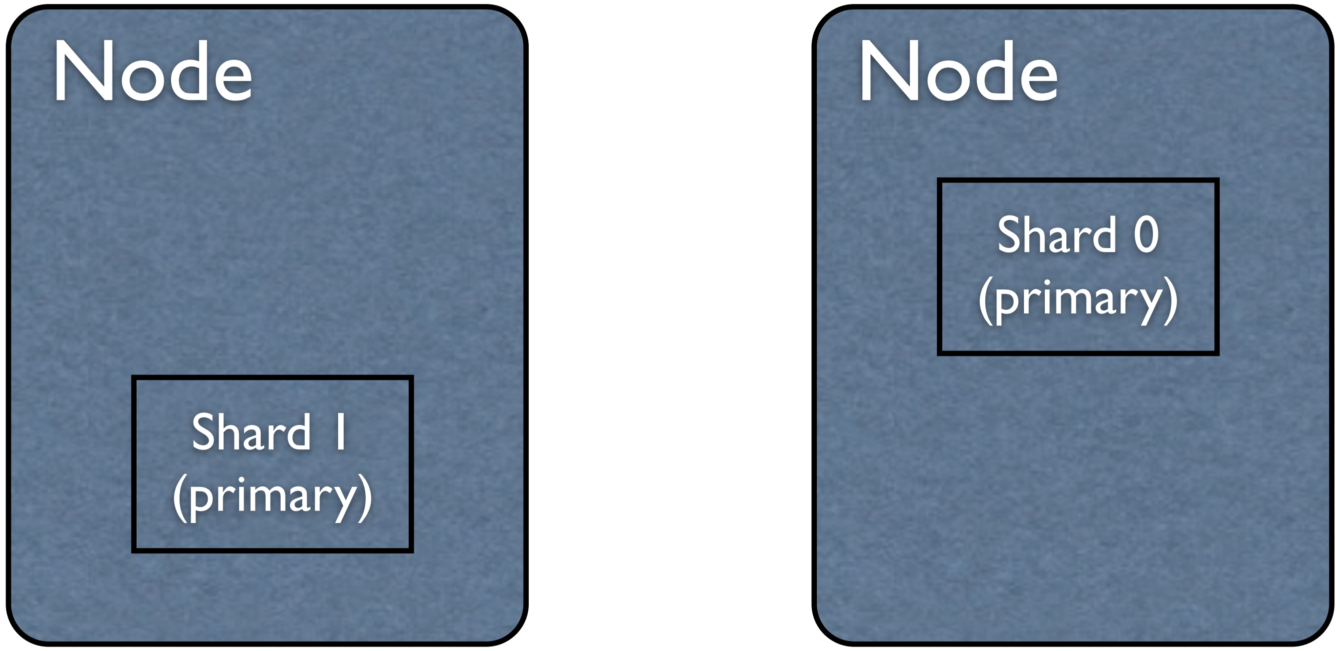


node failure



node failure - 1

- Replicas can automatically become primaries



Node

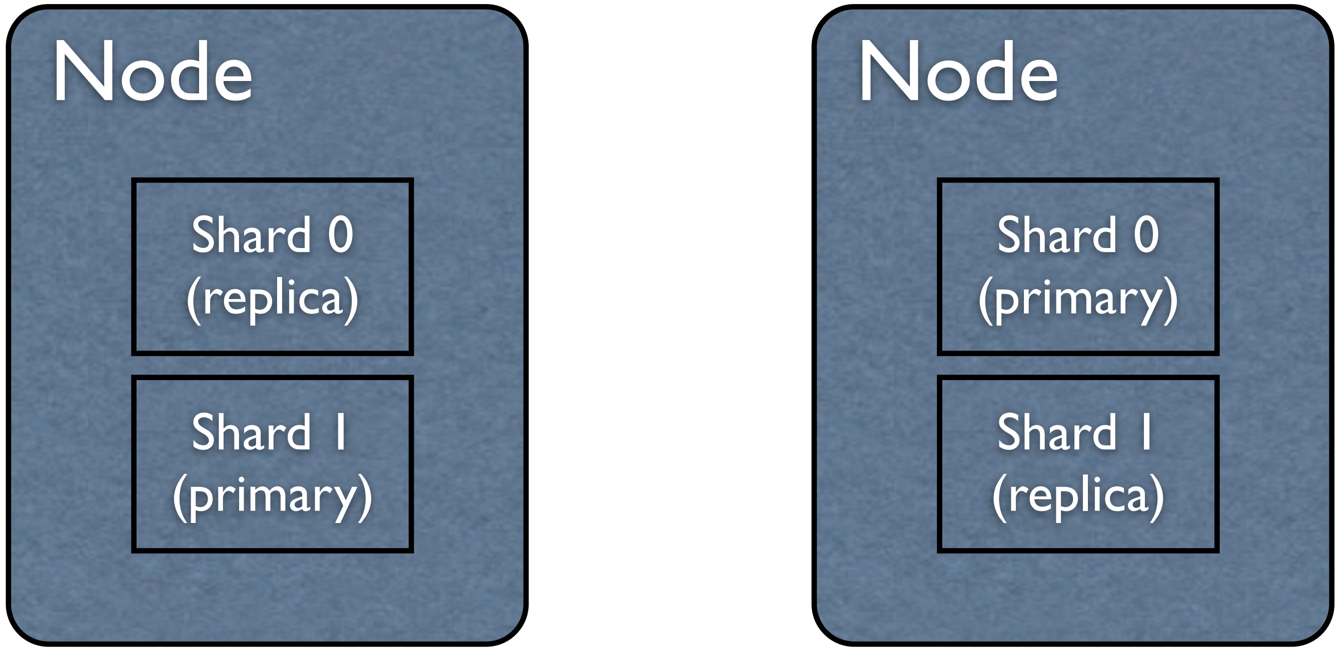
Shard 1 (primary)

Node

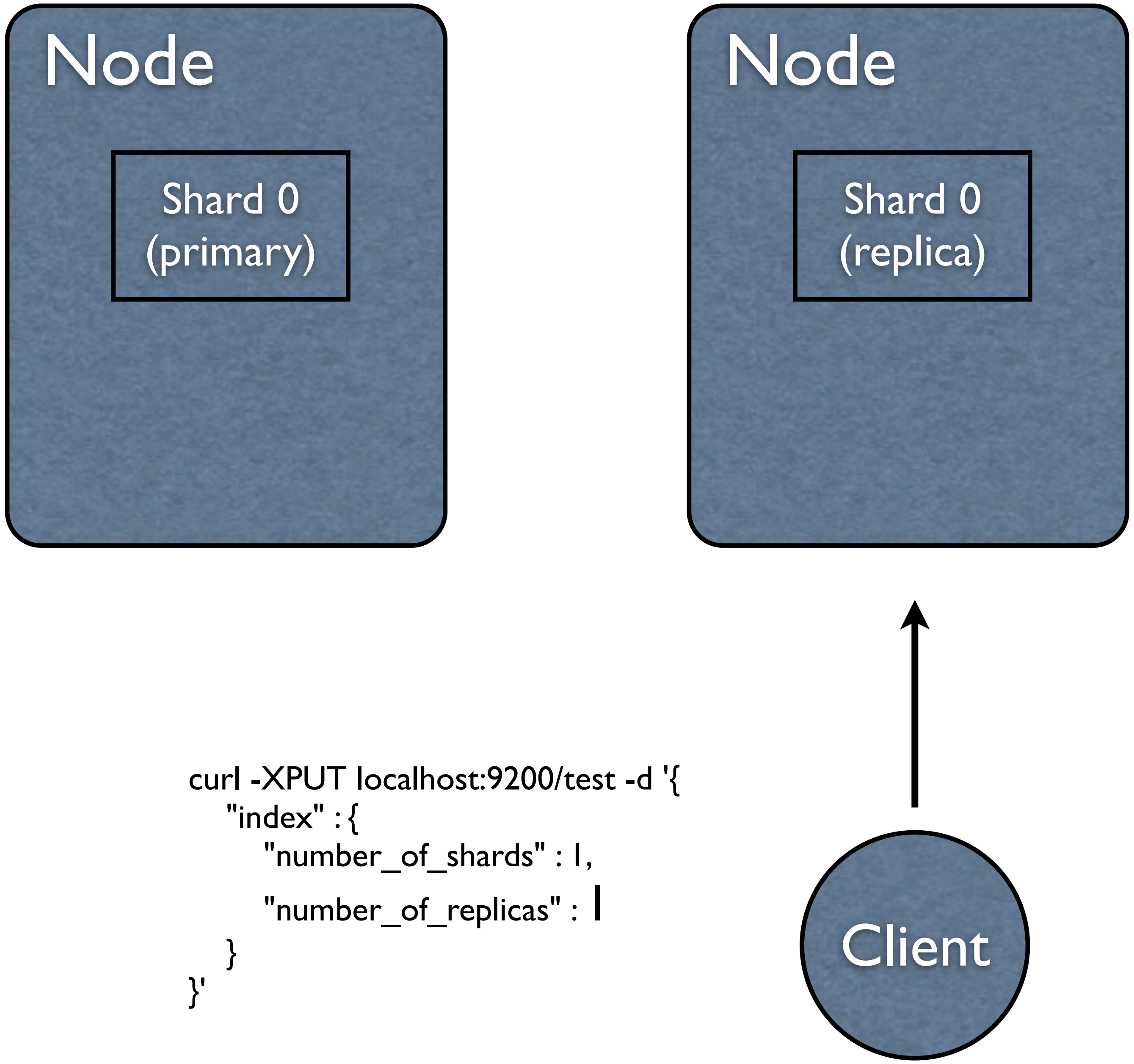
Shard 0 (primary)

node failure - 2

- Shards are automatically assigned, and do "hot" recovery

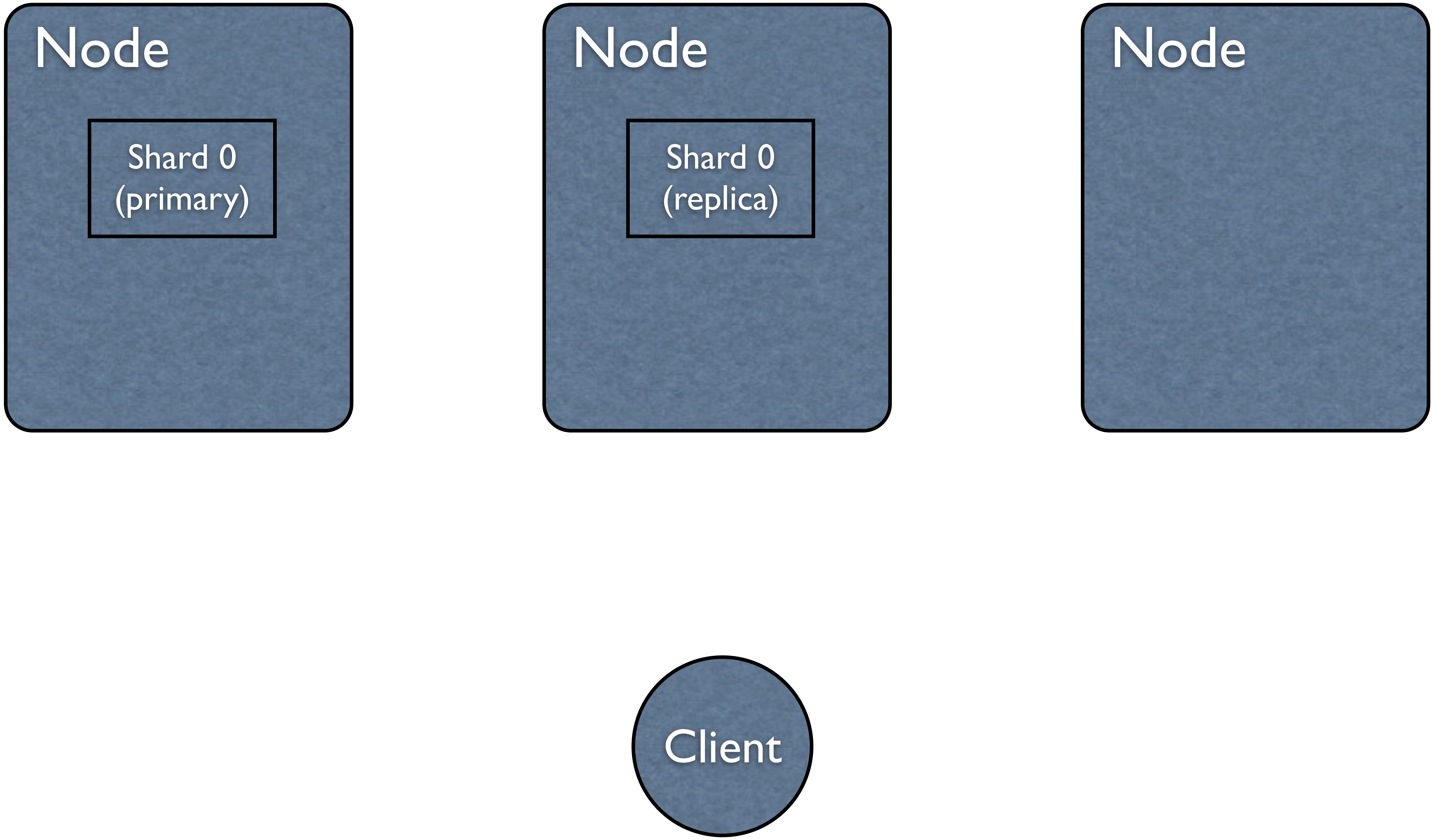


dynamic replicas

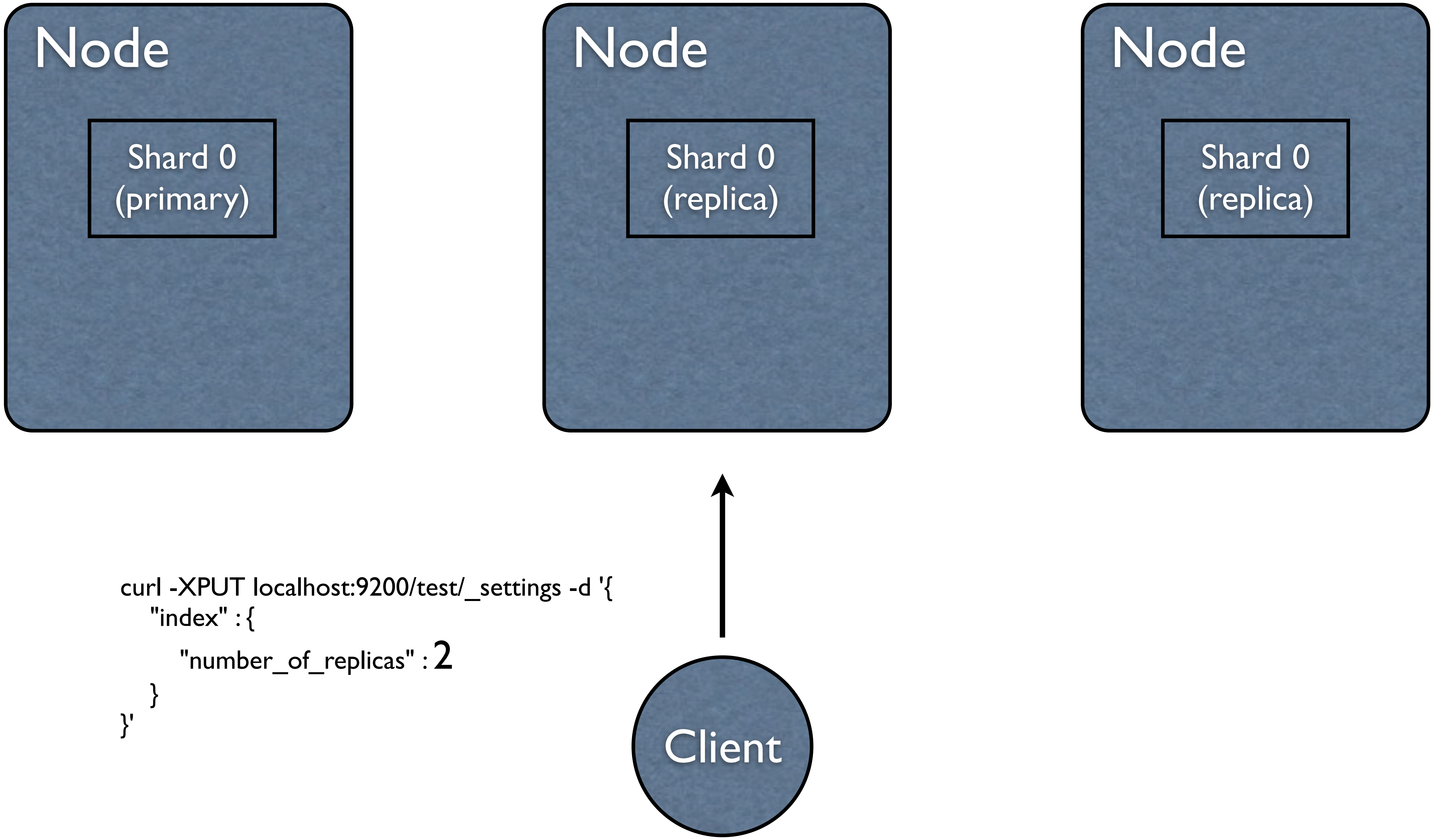


```
curl -XPUT localhost:9200/test -d '{
    "index" : {
        "number_of_shards" : 1,
        "number_of_replicas" : 1
    }
}'
```

dynamic replicas



dynamic replicas



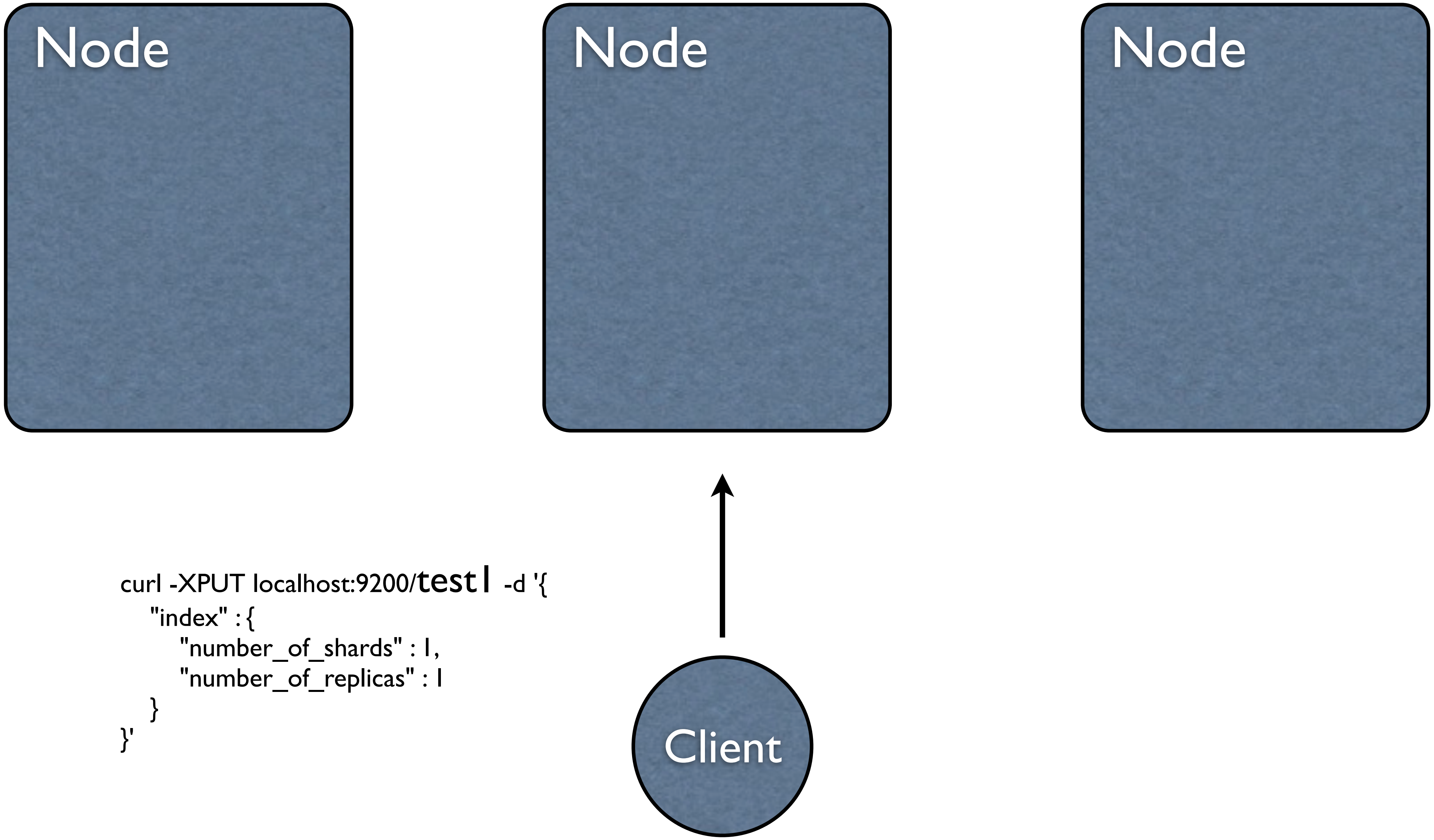
```
curl -XPUT localhost:9200/test/_settings -d '{
    "index" : {
        "number_of_replicas" : 2
    }
}'
```

transaction log

- Indexed / deleted doc is fully persistent
    - No need for a Lucene IndexWriter#commit
- Managed using a transaction log / WAL
- Full single node durability (kill dash 9)
- Utilized when doing hot relocation of shards
- Periodically "flushed" (calling IW#commit)

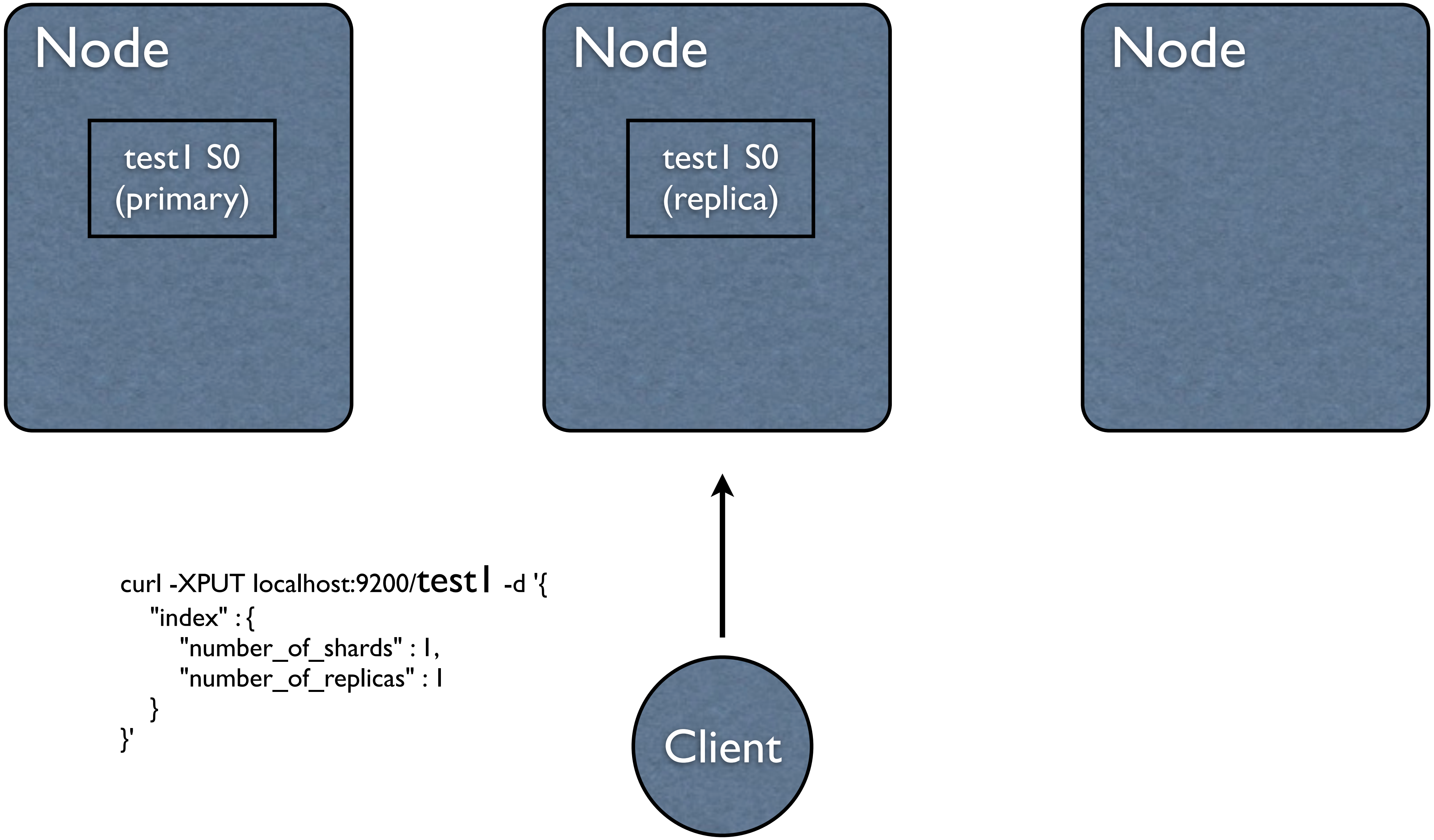
Multi Tenancy

multi tenancy - indices



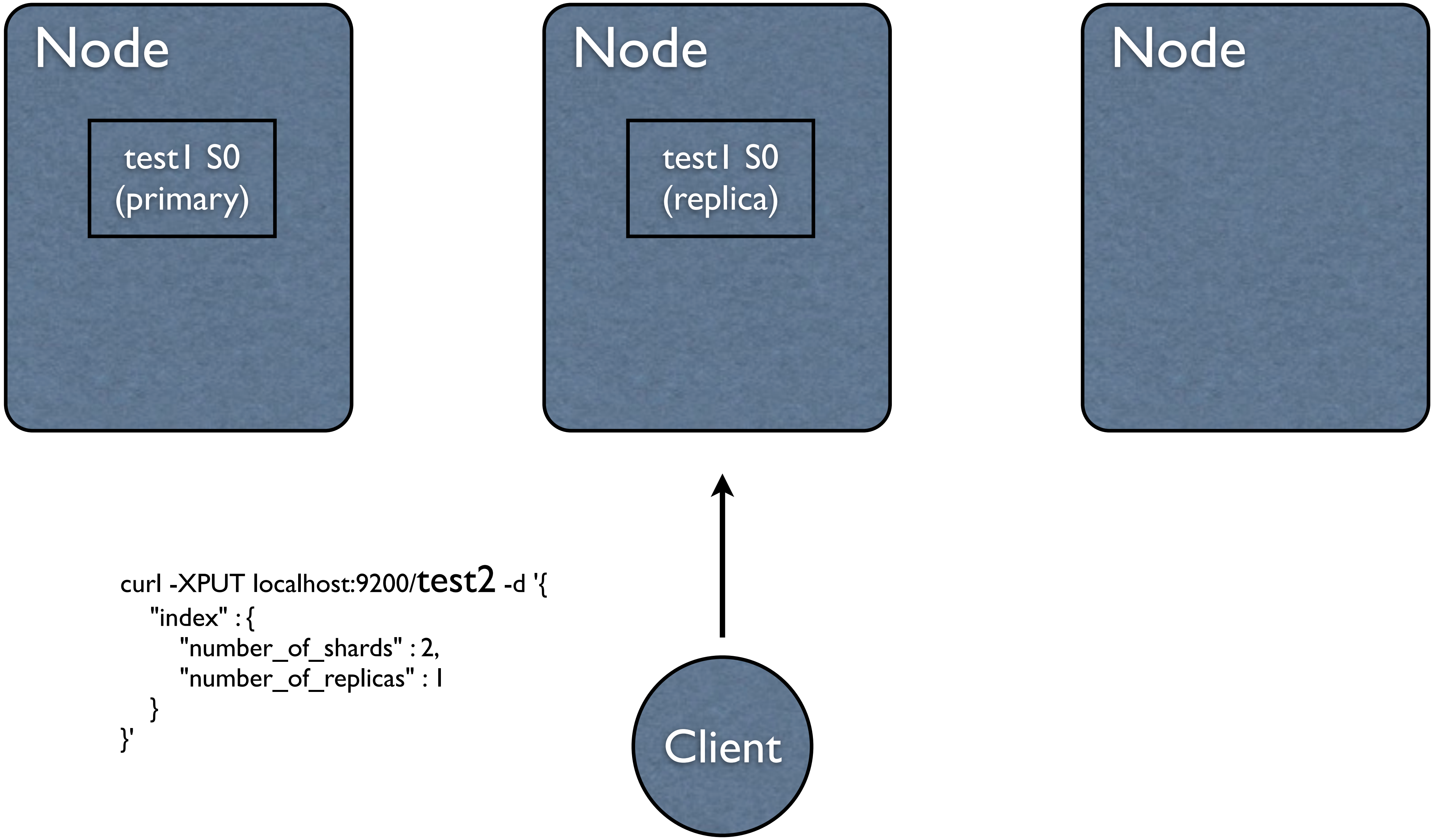
```
curl -XPUT localhost:9200/test1 -d '{
    "index" : {
        "number_of_shards" : 1,
        "number_of_replicas" : 1
    }
}'
```

multi tenancy - indices



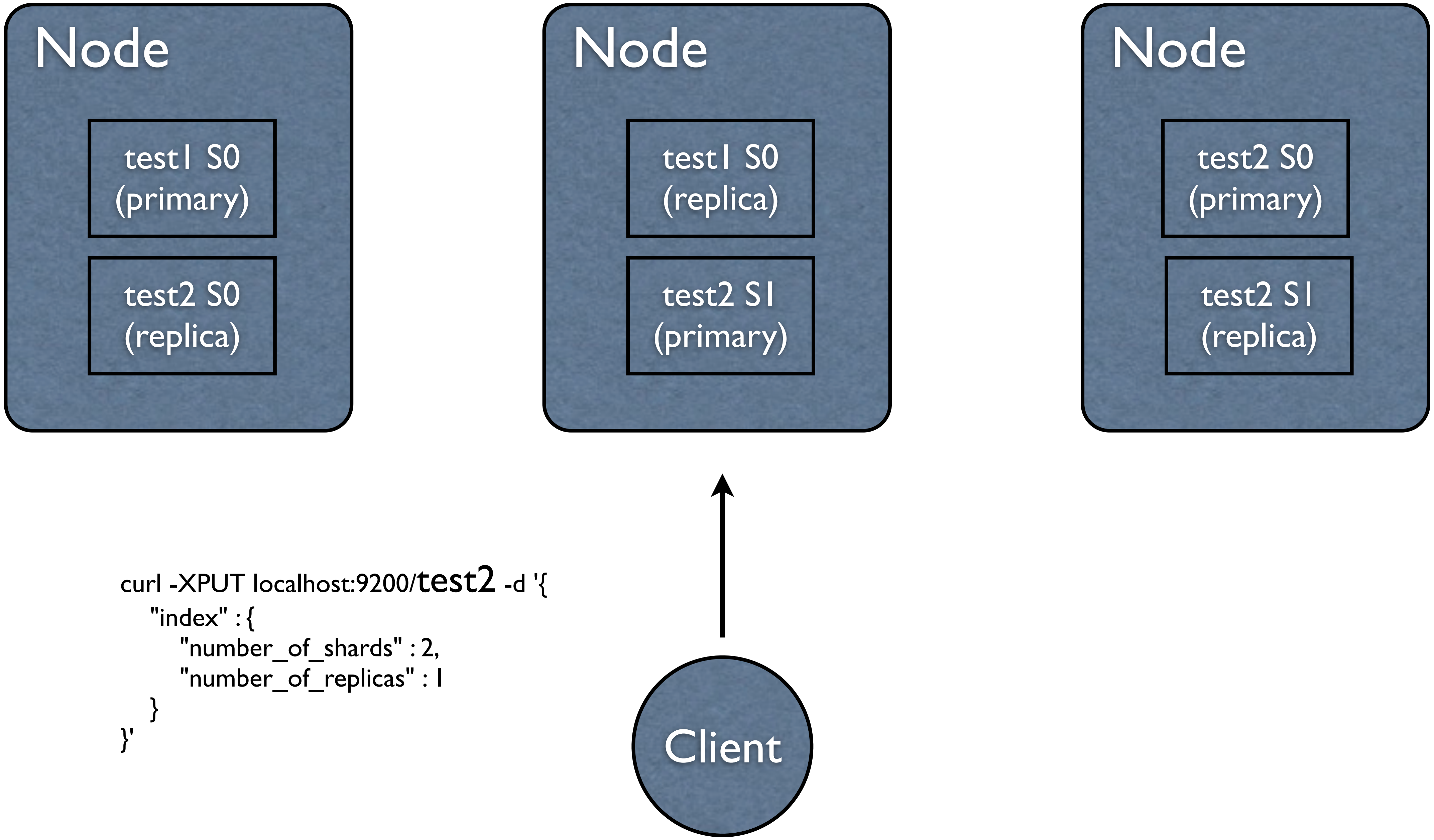
```
curl -XPUT localhost:9200/test1 -d '{
    "index" : {
        "number_of_shards" : 1,
        "number_of_replicas" : 1
    }
}'
```

multi tenancy - indices



```
curl -XPUT localhost:9200/test2 -d '{
    "index" : {
        "number_of_shards" : 2,
        "number_of_replicas" : 1
    }
}'
```

multi tenancy - indices



```
curl -XPUT localhost:9200/test2 -d '{
    "index" : {
        "number_of_shards" : 2,
        "number_of_replicas" : 1
    }
}'
```

multi tenancy - indices

- Search against specific index
  - curl localhost:9200/test1/_search
- Search against several indices
  - curl localhost:9200/test1,test2/_search
- Search across all indices
  - curl localhost:9200/_search
- Can be simplified using aliases

Applications

- Unstructured search functionality
  - typical free text query (text analysis)
- Structured search functionality
  - Query DSL (mainly Filters)
- Data Aggregation & Analytics
  - Facets (stats, histograms)
- Alerts
  - Percolation